Stereoscopic High Dynamic Range Video

Dominic Rüfenacht

Master Thesis

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne

Supervising Professor:
Sabine Süsstrunk

Supervisors at Philips:
Martin Hammer
Chris Varekamp

PHILIPS
sense and simplicity

August 19, 2011

Abstract

Stereoscopic video content is usually being created by using two or more cameras which are recording the same scene. Traditionally, those cameras have the exact same intrinsic camera parameters. In this project, the exposure times of the cameras differ, allowing to record different parts of the dynamic range of the scene. Image processing techniques are then used to enhance the dynamic range of the captured data.
A pipeline for the recording, processing, and displaying of *high dynamic range (HDR)* stereoscopic content, acquired using inexpensive *low dynamic range (LDR)* cameras, is proposed.
Two different approaches to obtain stereoscopic HDR content are presented and compared. In the *temporal approach*, different parts of the luminance range of the scene are recorded by temporally changing the exposure time of both cameras. Information from adjacent frames captured by the same camera is then used in order to increase the dynamic range. In the *spatial approach*, both cameras are assigned a distinct, fixed exposure time. Here, the dynamic range is increased by combining data from the cameras.
It is found that the intrinsic problems of the spatial approach are much more difficult to deal with than the ones of the temporal approach. In particular stereo matching, the critical component to combine data in the spatial approach, is more difficult than traditionally because the two cameras have different exposure times.
The results are evaluated for both static scenes and scenes with object movement using an objective quality metric of the visible differences of the stereoscopic pair independently, and visual evaluation on a stereoscopic display to evaluate the stereoscopic quality.

Contents

Glossary, Acronyms, and Notations

Glossary

exposure	One image of an exposure series.
exposure series	All images that are used to compute the HDR radiance map.
foreground fattening	Effect arising commonly in block-based disparity estimation for blocks containing objects at different depths. The pixels belonging to the object that is further away are mistakenly assigned the same disparity as the ones of the object in the foreground.
fronto-parallel surface	A surface that is directly facing the cameras, opposed to a slanted surface.
gamut	Subset of colors that can be accurately represented by a device.
luminance	Photometric measure of the luminous intensity per unit area of light going in a given direction. It is measured in candela per meter squared [cd/m^2].
re-exposure	Simulating a longer/shorter exposure time. Due to clipped regions and noise, it is never possible to get a perfect reconstruction of a given exposure time.
semaphore	Abstract data type which can be used to limit the number of threads accessing a common resource in a concurrent programming environment.

Acronyms

CCD	charge-coupled device
CRF	camera response function

EXIF	exchangeable image file format
HDR	high dynamic range
HVS	human visual system
IDP	iterative disparity propagation
IPL	inner plexiform layer
LDR	low dynamic range
LUT	lookup table
MS-SSIM	multi-scale structural similarity
MSE	mean squared error
NCC	normalized cross correlation
OPL	outer plexiform layer
PSNR	peak signal to noise ratio
SAD	sum of absolute differences
SNR	signal to noise ratio
SSIM	structural similarity
VDP	visible difference predictor
WTA	winner takes it all

Notations

k	$k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$, where R, G, B refer to the red, green, and blue color channel, respectively.
e	Exposure ratio, defined as $e = \frac{\Delta t_2}{\Delta t_1}$, where $\Delta t_2 \geq \Delta t_1$.
$\mathrm{I}_{c_j,f}^{\Delta t_i}$	Image number $f \in \{1, 2, ..., N_{\mathrm{fr}}\}$, taken by camera $j \in \{1, 2, ..., N_{\mathrm{cam}}\}$, at an exposure time of $i \in \{1, 2, ..., N_{\mathrm{exp}}\}$.
$\hat{\mathrm{I}}_{c_j,f}^{\Delta t_i}$	Image number $f \in \{1, 2, ..., N_{\mathrm{fr}}\}$, synthesized to match view of camera $j \in \{1, 2, ..., N_{\mathrm{cam}}\}$, with an exposure of $i \in \{1, 2, ..., N_{\mathrm{exp}}\}$.
$\tilde{\mathrm{I}}_{c_j,f}^{\Delta t_i}$	Image number $f \in \{1, 2, ..., N_{\mathrm{fr}}\}$ taken by camera $j \in \{1, 2, ..., N_{\mathrm{cam}}\}$, with a simulated exposure of $i \in \{1, 2, ..., N_{\mathrm{exp}}\}$.

N_{cam}	Number of cameras.
N_{exp}	Number of different exposure times.
N_{fr}	Number of frames captured per camera.
N_{reg}	Number of regions in the region partition of an image.
$N_{\text{reg\_types}}$	Number of types of regions.
Ω	The complete set of regions. $\Omega = \bigcup_{i=1}^{N_{\text{reg\_types}}} \Omega_i$.
Ω_i	Subset i of regions. $\Omega_i \subseteq \Omega$.
$\mathrm{I}_{c_j}^{\Delta t_i}(x, y, k)$	Pixel at coordinates (x, y) of color channel k of image I, of exposure time $i \in \{1, 2, ..., N_{\text{exp}}\}$.
$S_{c_j,s}$	Segment s of camera j.
$S_{c_j}^{\Delta t_i}$	Segments of camera j, segmented on exposure time Δt_i.
Δt_i	Exposure time. By convention, $\Delta t_l \leq \Delta t_m, \forall l < m$.
Δt_{c_j}	Exposure time of camera j, measured in ms.
U	V $\bigcup$ W.
$\vec{u}_n^{(c_j,c_l)}$	Disparity vector either coming from V or W, going from the reference of camera j to camera l.
V	Set of disparity vectors found by region-matching.
$\vec{v}_n^{(c_j,c_l)}$	Disparity vector of region n, going from the reference of camera j to camera l.
W	Set of disparity vectors found for clipped regions by averaging over adjacent disparity vectors from V.
$\vec{w}_n^{(c_j,c_l)}$	Estimated disparity vector for a clipped region n based on neighboring disparity vectors from V, going from the reference of camera j to camera l.

List of Figures and Tables

List of Figures

List of Tables

1

Introduction

Imagine yourself on a sunny day hiking in the mountains. The view of the close-by mountain ridge is splendid, and you want to capture this moment with your camera. Unfortunately, the captured image does not match what you have seen. The nice sky is washed out and the lovely chalet in the foreground is much darker than you perceived it. Why is this happening? Key to understand this problem is the dynamic range of a scene, which is defined as the ratio between the highest and the lowest scene luminance value. The luminance range of the real world ranges from 10^{-4} cd/m^2 (starlight) up to around 10^9 cd/m^2 (direct sunlight), corresponding to 14 orders of magnitude. The *human visual system (HVS)* is capable of perceiving around 15 orders of magnitude, whereof 3 to 5 orders of magnitude can be perceived simultaneously using *local adaptation*. This is more than enough considering the fact that a typical natural scene has a contrast ratio lower than 10,000:1 [16]. Figure 1.1 shows approximate luminance values for typical scenes.

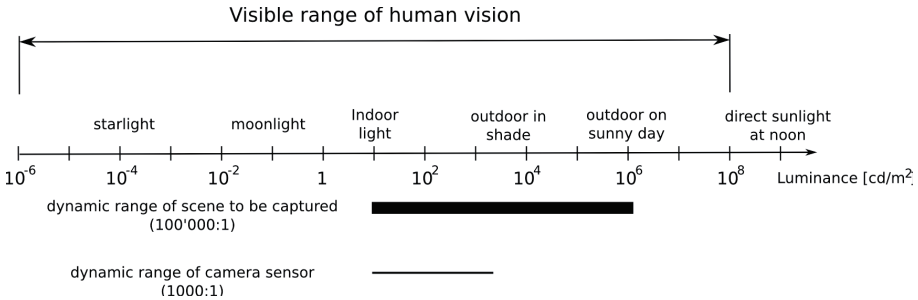


Figure 1.1: Approximate luminance values for different scenes. Based on data from [16].

This gives some indication about the dynamic range of a scene to be captured. Consider a scene showing the interior of a house with a window on a sunny day. One can easily estimate the dynamic range of this scene using the approximate luminance values of the figure above. For the scene mentioned, one can read the luminance value for an outdoor scene (10^6 cd/m^2), and divide it by the luminance value for indoor light (10 cd/m^2). This results in an approximate dynamic range of 100,000:1. A normal camera sensor is able to capture a dynamic range of only around 1,000:1. This implies that if the scene has a higher dynamic range, the sensor will not be able to capture it, and the resulting image

will be clipped in bright and/or dark parts.
While there exist a few HDR sensors that are able to capture a greater dynamic range, such as the OmniPixel3-HSTM™from Omnivision [28], these are reserved for professional sectors due to their very high price tag. The most common technique to increase the dynamic range which does not involve changing the camera sensor itself, is to take several differently exposed images of the same scene and combine their information in order to get information in both the dark and the bright regions of the captured scene [33]. One example of such an exposure series is shown in Figure 1.2.

(a) $\Delta t_1 = \frac{1}{500}$s (b) $\Delta t_2 = \frac{1}{125}$s (c) $\Delta t_3 = \frac{1}{30}$s (d) Tone mapped HDR image

Figure 1.2: Same scene captured with different exposure times, and the resulting tone mapped HDR image.

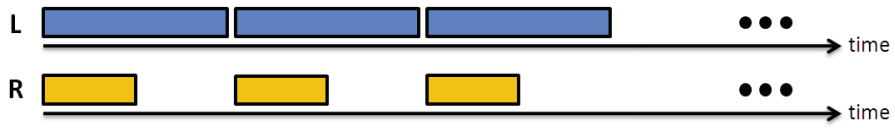
We can see that in the images with a short exposure time the camera captured the details in the bright areas of the image, and in the images with long exposure time details in the dark areas of the scene are visible. For this method to work properly, the images to be combined together need to be correctly aligned. In the end, the result needs to be tone mapped to the displayable gamut of the viewing device.
Another limitation of camera sensors is that by taking an image of a real scene with a camera, the 3D scene is projected onto a 2D image sensor, and the depth information is lost. Humans perceive depth due to several cues, an important one being *stereopsis*, i.e. the fact that the left and the right eye get information from slightly different viewpoints. Recording a scene with two cameras which are on the same horizontal baseline with identical specifications and settings allows to record the information for the left and the right eye at the same time. When these two separate streams of images are fed to the left and the right eye using a stereoscopic display, we perceive the recorded scene in 3D.

Stereoscopic TVs have become popular over the last two years. In the future, TVs will be capable of displaying HDR stereo content. This creates the need to be able to record *stereoscopic HDR video*, i.e stereo video content with a higher dynamic range. While one could use two HDR cameras in order to record stereoscopic HDR video, this solution would be too expensive for the consumer market. There is therefore an interest in a low-cost solution that is able to record high dynamic range stereo content using inexpensive LDR cameras.

1.1 Problem Definition

This thesis aims at investigating the possibility of recording *stereoscopic HDR video* using LDR cameras. On one side, traditional stereo setups require the two cameras to have the same intrinsic parameters, and to take images from slightly different points of view. In particular, this means that the exposure time is the same for the two cameras. On the other side, HDR imaging needs the exposures to capture different parts of the dynamic range by changing the exposure time of the captures. A second requirement is that the exposures need to be aligned. By combining stereo with HDR, we need to violate those fundamental assumptions. In other words, we try to combine two domains that seem to be mutually exclusive. We investigate two different modes to record *stereoscopic HDR video* (see Figure 1.3), which in turns influence the way they are processed.

(a) Temporal Stereoscopic HDR



(b) Spatial Stereoscopic HDR

Figure 1.3: Recording modes investigated in this thesis for the R(ight) and L(eft) camera. The temporal mode varies short and long exposure times for both cameras, whereas the short and long exposure are fixed to one camera in the spatial mode.

Due to the nature of how the stereoscopic content is created, we call the two modes *spatial* and *temporal stereoscopic HDR*. The spatial mode sets a fixed, distinct exposure time to the two cameras. The temporal mode temporally varies between short and long exposure times for both cameras. The two main questions we want to get answered are which recording mode gives the best results and by how much the exposure times can differ to still get good, consistent stereo pairs. For this, a pipeline for creating stereoscopic HDR content is proposed, which includes recording, processing, and evaluation of stereo HDR content.

1.2 Related Work

The literature review is organized in three parts. First, HDR image and video creation which is needed to increase the dynamic range of the captured scenes is discussed. In order to align the images, stereo matching is used, which is the focus of the second part. In the last part, work aiming at combining HDR imaging and multi-view matching is presented.

1.2.1 HDR Imaging

Computational photography is a domain of research where image processing and analysis algorithms are applied on one or several images in order to create images that go beyond the capabilities of the imaging device. One instance of this domain is HDR imaging, where several images with different exposure times are combined together with the aim of creating an image with a dynamic range that goes beyond the dynamic range of the sensor of the capturing device.
In order for the HDR algorithms to work correctly, the response of the camera needs to be linearized. This means that the intensity measured by the sensor is linearly related to the real scene radiance. Debevec and Malik [5] have proposed a method for estimating the camera response function, which relates digital values to scene radiance, by taking several images with different known exposure times of the same static scene. It is based on the observation that while the measured brightness values will change between the different images, the scene radiance value will stay constant. This allows them to estimate the inverse of the camera response function without knowing the real scene radiance.
Once the camera response function f has been estimated, the images can be linearized by applying the inverse of the camera response function $g = f^{-1}$.
The next step is to estimate the radiance map of the real scene. The general form of the function used to recover the radiance values is in all methods based on a weighted average. Different weighting functions accounting for the reliability of a pixel measurement have been proposed. Debevec and Malik [5] use the same weighting function as they use to estimate the camera response function. As mentioned before, this hat function gives a higher weight to exposures where the pixel's value is closer to the middle of the response function. Mann and Picard [20] observe that the flatter regions of the curve contain less information about the incoming radiance, and use the derivative of the inverse camera response function to account for that. In order to make the quantization error perceptually uniform, they compute the derivatives on a logarithmic scale, since the higher the pixel intensity value is, the less sensitive to changes in intensity value the HVS is. This phenomenon can be described by the *Weber-Fechner law*. Mitsunaga and Nayar [24] show that to maximize the *signal to noise ratio (SNR)*, the weighting function needs to emphasize both higher pixel values and larger gradients in the response function. They consider the camera noise unknown and assume that the uncertainty is constant across the output range. In order to suppress over- and under-exposed values, Reinhard *et al.* [33] add a hat function to the weighting function proposed by Mitsunaga and Nayar. This makes sense, since those outliers would only distort the result. Tsin *et al.* [41] use the camera noise model by applying a weighting function that is based on the output standard deviation. In [10], Granados *et al.* show that under the assumption of compound-Gaussian noise, the standard deviation is suboptimal. They adopt a camera noise model that accounts for both temporal and spatial noise sources. Calibrating the sensor parameters beforehand allows them to iteratively estimate the irradiance and its uncertainty in a way that noisy measurements do not influence the weighting function. They define an optimal compound-Gaussian model that takes all noise sources involved in the image capturing process into account, i.e. temporal (photon and dark current shot noise, readout noise) and spatial (photo-response and dark current non-uniformity). Their weighting function is then derived based on this model and applies to linear sensors.

Their method outperforms all the other approaches in terms of SNR. However, they do not report on the computational complexity of their algorithm.

The fact that these methods depend on the assumption that the exposure time is known does not constrain them, since exposure times can easily be retrieved from the *exchangeable image file format (EXIF)* data, or in development settings it can be saved during the capture. A more severe requirement is that the images in the exposure series are aligned. This requirement is often not satisfied due to motion in the scene created by moving objects or by changing the camera position. The problem of motion between images of an exposure series has been addressed in different ways. The easiest motion to correct is the one when the camera was rotated, but the scene remained static.
Tomaszewska and Mantiuk [39] present a fully automatic method for eliminating misalignments between a sequence of hand-held photographs that were taken at different exposure times. Their method uses the SIFT-method to find key-points between the different images, and then a homography is applied to warp the images to a reference image. While they claim that their method works for misalignments caused by any movement of the camera, the fact that they use a homography to align the images implies that it will not work for translational movements of the camera. Also, they do not account for moving objects in the scene. Jacobs *et al.* [14] propose a fully automatic way to align a set of LDR images while removing the influence of both camera movement and moving objects in the final HDR image. They propose two modules that can be integrated into the normal HDR image generation. The first module realigns images that were taken with a camera that has undergone small rotations and translations that are typical for handheld camera pictures. They remark that their method fails if moving objects occupy a too large part of the scene. Images that have small misalignments due to camera shake can be effectively aligned, provided that there is no or only small motion in the scene.
More often than not, the scene to be captured contains objects that are in motion. This results in an exposure series where some pixels in the different exposures do not belong to the same object in the real scene. If this motion is not accounted for, the radiance map will be distorted. Due to the fact that moving objects will be averaged with the object in the back, they will look as if they were semi-transparent, which is why they are called *ghost-artifacts*. Jacobs *et al.* [14] propose a way to remove the effect of moving objects in order to avoid ghost-artifacts, and propose two different ways to detect motion in a scene. The first one is based on variance, by noting that pixels which are affected by movement generally have a larger irradiance variation over the different exposures. The second method to remove object movement is based on a statistical, contrast-independent measure based on the concept of entropy. This approach is similar to the ones used by Jing *et al.* [15] and Ma and Zhang [18], but can be used on images taken under different illumination and with varying exposures.

Kang *et al.* [17] create HDR video from an image sequence of a dynamic scene by rapidly varying the exposure time of each frame. Their workflow consists of three main steps. First, they use automatic exposure control during capture by using a real-time exposure control. Their approach can be seen as a subsampling in the temporal dimension. Once the acquisition is done, the captured frames are motion-compensated and a full radiance map is estimated for each frame. This way, dense correspondences between frames are

obtained in order to combine the pixels from different exposures. The last step consists in tone mapping the generated HDR images. In order to avoid temporal inconsistencies, statistics from neighboring frames are used. Their implementation does not handle occlusions, and also encounters problems when there are too many non-rigid effects such as specularities and inter-reflections in the scene. Mangiat and Gibson [19] build upon the work of Kang *et al.* [17], but use a simpler block-based motion estimation. They enhance the results using the color information from areas with good correspondences and edge information of the current frame to detect and correct poorly registered pixels. They use the hat function proposed by Debevec and Malik [5], with some additional constraints. Due to the simple weighting function, block artifacts and other misregistrations of pixels are passed on to the tone mapping. Using a cross-bilateral filter, they combine the color information of the HDR image with edge information of the current frame, which removes block artifacts of pixels that can be predicted to be erroneous. It fails, however, for general block artifact removal in saturated regions.

When it comes to evaluating the perceived quality of an HDR image, two popular ways can be followed. *Subjective quality evaluation methods* based on psychophysical evaluation require many participants in order to provide reliable results, which makes them time-consuming, and often also expensive. *Objective quality metrics* are less reliable than subjective ones, but the fact that they are completely automatic makes them the method of choice in many situations. Mantiuk *et al.* [21] propose an objective quality metric to predict visible differences in HDR images based on models of the HVS. It computes a map of probability values indicating how likely differences between a reference HDR and the produced HDR image will be perceived by a human observer. Using a pooling function, the prediction of visible differences is pooled to a single value predicting the perceived quality of an image.

1.2.2 Stereo Matching

One of the most investigated topics in computer vision is stereo matching. For this project, we are interested in dense correspondence maps. With the aim of bringing some order into the jungle of different dense correspondence algorithms proposed in literature, Scharstein and Szeliski [35] developed a taxonomy that allows assessing different components in stereo methods. One major outcome of their work is the *Middlebury test set* [1], where a *test bed* for the quantitative evaluation of stereo algorithms can be found. One can also find a ranking comparing over 100 different dense stereo matching algorithms. The ranking is solely based on disparity map correctness, and completely disregards computational complexity. As with all test beds, the results do not indicate how well an algorithm generalizes.
Stereo correspondence algorithms can be roughly classified in two main categories, namely *local* and *global* methods. While local algorithms compute the disparity of a pixel by only looking at a finite window around that pixel to aggregate a matching cost, global algorithms minimize an energy function that is based on the whole image [37]. Brown *et al.* [2] give a good overview and comparison over the main algorithms used in local and global stereo matching. Local methods are computationally less expensive than global ones, but

[1] `http://vision.middlebury.edu/stereo/`

have problems in homogeneous regions and in (half-)occluded areas of the image, as the matching results are ambiguous. Global correspondence methods are better at handling these problematic regions because they use non-local constraints [37].
Based on the reasonable assumption that neighboring pixels with similar colors have similar depths, color-based image segmentation has been used in order to simplify the stereo matching problem. Several similar pixels that are grouped together are often referred to as *superpixels*. Using superpixels not only reduces the problem of depth-discontinuity at object boundaries, but can also greatly reduce the computational complexity. Because of these interesting properties, we chose to use a segmentation-based disparity estimation method for this project.
Superpixels have been used in combination with both local and global methods. Birchfield and Tomasi [1] first oversegment the image and then match the segmented regions. Instead of relying on the *fronto-parallel surface assumption*, they use an affine model in order to account for slanted surfaces. Zitnick and Kang [44] oversegment the image, and then apply loopy belief propagation where the message passing is done between segments rather than single pixels. In combination with a self-adapting matching score and a robust plane-fitting technique, they got very good rankings on the Middlebury test set. Hong and Chen [11] use a segment-based approach in combination with graph cuts. Once the image is segmented, graph cuts finds the optimal labelling in a matter of seconds, which is much faster than if it is applied on a pixel basis.

1.2.3 Multi-View HDR Imaging

There are only few attempts to create multi-view HDR content. To our knowledge, Troccoli *et al.* [40] were the first to propose a technique that adapts multi-view stereo to images taken at different exposure times, in order to simultaneously recover depth maps and HDR textures. They assume static scenes, and use a different exposure time for every camera(spatial approach). They show that under the assumption that the camera has a gamma response function and ignoring noise and quantization effects, the normalized cross-correlation (NCC) is exposure invariant. They obtain a depth map and an HDR image for each camera. Unfortunately, their evaluation only shows some sample depth maps, which makes it difficult to assess both the quality of the depth maps and of the HDR images. Also, the requirement of static scenes greatly simplifies the task, and makes this approach unsuited for video.
Sun *et al.* [25] improve on [40] by using graph cuts to do the disparity estimation, and by adding a disparity refinement stage. This results in HDR images with fewer artifacts and allows to encode a larger dynamic range. They do, however, not report on computational time. The fact that they use graph cuts for both the initial disparity map estimation and in the refinement stage suggests that computational complexity is high. It is mentioned that thanks to the fact that the left and right view are captured at the same time instance, the system is able to capture scenes with fast motion. While this is true, no evaluation of the temporal stability has been reported.
The work of both Troccoli *et al.* [25] and Sun *et al.* [40] was focused on disparity estimation, exploiting the higher dynamic range to find better matches. Also, both approaches have only been tested on images from the Middlebury test set and on computer generated graphics. All the test images used in the two papers do not contain any

regions of saturated pixels, which favours the assumption that the focus was on disparity estimation rather than increasing the dynamic range.
Ramachandra *et al.* [31] use Kang *et al.*'s [17] method of processing HDR video and apply it to multiple cameras, using information from the other cameras as well as from adjacent frames taken by the same camera. While it is clear how the images are aligned in the temporal domain, it is not mentioned how information is exchanged between the different cameras (spatial approach). They claim that the results are better for their proposed method, however there is no quantification of the results, and the provided images look all very similar. Unfortunately there is no information on how the image alignment in the spatial mode is done. Their main focus is on deblurring long exposures which are subject to motion blur.

1.3 Overview of the Thesis

This work differs from previous work in that the whole stereoscopic HDR pipeline is implemented and tested. This allows the revelation of problems that are not necessarily obvious if only one part of the pipeline is investigated. We limit the practical implementations on two cameras. While [40] and [25] used the HDR radiance maps to create better disparity maps, our work is focused on the quality of the HDR radiance maps. In particular, we are not interested in creating new views (view interpolation), i.e. the quality of the disparity maps is secondary. We further only allow object movements in the scene, and assume that the longest exposure time is short enough that no motion blur occurs. Since there is no stereoscopic HDR display available, the stereoscopic HDR video part of the evaluation will be carried out on tone mapped HDR images. The rest of the evaluation will be carried out on the HDR radiance maps.

Figure 1.4: Main parts of the stereoscopic HDR video pipeline.

Figure 1.4 shows the three main parts of the proposed stereoscopic HDR pipeline. First, the frames are captured and stored on the hard drive. They are then processed in order to generate a stereo pair with increased dynamic range. In the last part, the resulting (stereo) HDR radiance maps for both still scenes and video are evaluated.
The pipeline is reflected in the structure of the report. Each chapter is devoted to one part of the stereoscopic HDR pipeline.
In **Chapter 2**, the experimental setup used to record the scenes is shown. The temporal and spatial stereoscopic HDR modes are presented, and then it is explained how camera synchronization has been achieved.
The processing of the raw image data is explained in **Chapter 3**. Processing includes the estimation of the camera response function, image alignment, and HDR radiance map estimation. The tone mapping used in order to show the HDR image on a display with lower dynamic range is explained at the end of Chapter 3.
In **Chapter 4**, three objective quality metrics are compared. The one that is best suited

for our purposes is then used to evaluate the HDR radiance maps obtained in the processing part. Several tests are performed to find a minimal number of exposures. The quality of the stereoscopic HDR image pairs for still scenes and video are evaluated by visual inspection on a stereoscopic display.
Conclusions are found in **Chapter 5**, and directions for future work are proposed.

1.3.1 Notations

All notations used in this report are defined in the notations section. This section highlights the most important notations which will be used throughout the report. Let us denote N_{cam} the number of cameras, N_{exp} the number of different exposure times, and N_{fr} the number of frames taken by each of the cameras. We then denote $\mathrm{I}_{c_j,f}^{\Delta t_i}$ an image taken by camera $j \in \{1, 2, ..., N_{\text{cam}}\}$, $i \in \{1, 2, ..., N_{\text{exp}}\}$, and $f \in \{1, ..., N_{\text{fr}}\}$. In case of still scenes, f is omitted and the notation simplifies to $\mathrm{I}_{c_j}^{\Delta t_i}$.
We further denote a specific pixel of camera j of color channel k, taken at exposure time i, as $\mathrm{I}_{c_j}^{\Delta t_i}(x, y, k)$, where $k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ and i as above. Note the distinction between an image and a pixel. As soon as an image is indexed, we refer to a specific pixel of that image. The index for the exposure time may be replaced by HDR, meaning that the image is composed of several exposures.

2

Recording

The first part of the proposed stereoscopic HDR pipeline is the recording of a scene. Two ways of capturing stereoscopic HDR video with LDR cameras are presented in this chapter. In the first one, the exposure time of each camera is changed separately, which we further refer to as *temporal stereoscopic HDR* or *temporal approach*. The second one is to set one camera to a short exposure time, and the other to a long(er) exposure time, referred to as *spatial stereoscopic HDR* or *spatial approach*.
We start with a detailed description of the two proposed recording modes, followed by a description of all the components used in the setup.

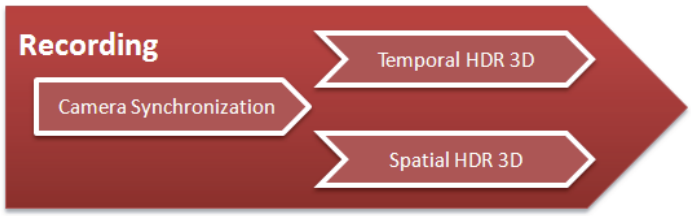


Figure 2.1: Parts of the recording block.

2.1 Temporal Stereoscopic HDR

In this mode, the left and the right camera independently record the raw data for the HDR video by varying the exposure time for each frame. Figure 2.2 shows the basic idea of the setup for two different exposure times.
For both cameras, the exposure time is changed after each capture. Let us assume that we change the exposure time from shortest to longest, and that there are N_{exp} different exposure times. For camera c_j, we therefore record the following sequence: $\mathrm{I}_{c_j,1}^{\Delta t_1}, \mathrm{I}_{c_j,2}^{\Delta t_2}, ..., \mathrm{I}_{c_j,N_{\text{exp}}}^{\Delta t_{N_{\text{exp}}}}, \mathrm{I}_{c_j,N_{\text{exp}}+1}^{\Delta t_1}, ..., \mathrm{I}_{c_j,F}^{\Delta t_{(F-1) \bmod (N_{\text{exp}})+1}}$. For every frame f, we create the HDR image $\mathrm{I}_{c_j,f}^{\text{HDR}}$ by using the actual frame and the $N_{\text{exp}} - 1$ future frames, which allows us to keep the original framerate. It is clear that one can choose $N_{\text{exp}} > 2$ exposures for the HDR sequence, allowing to capture a higher dynamic range at the price of a reduced capability of capturing fast motion.

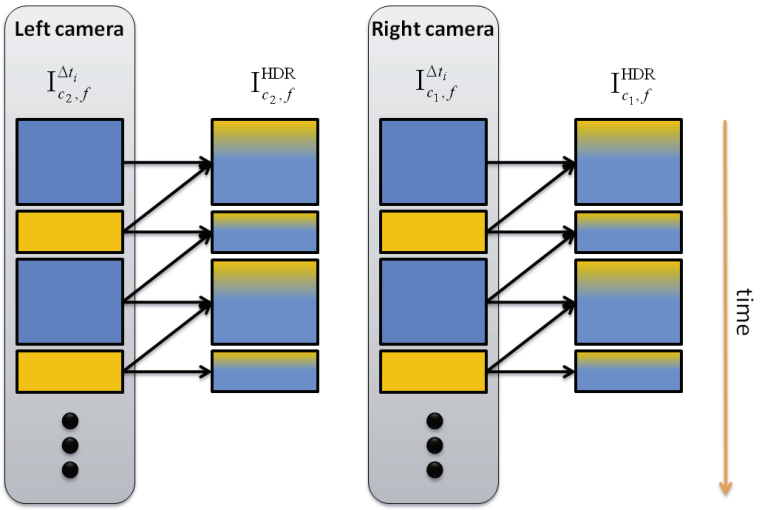


Figure 2.2: The temporal stereoscopic HDR captures the left and the right view independently. Here, $N_{\text{exp}} = 2$.

2.1.1 Theoretical Bound on Capturing Time for an Exposure Series

One interesting aspect of the temporal HDR is the following observation. Let us assume that we take N_{exp} different exposure times $\Delta t_1 < \Delta t_2 < ... < \Delta t_{N_{\text{exp}}}$, and let $\Delta t_1 = a$, where a is a constant. If we vary the exposure time by powers of 2, i.e. $\Delta t_{i+1} = 2^i \Delta t_1, \forall i \in [1, N_{\text{exp}} - 1]$, the following equation holds:

$$\lim_{N_{\text{exp}}\to\infty} \sum_{i=1}^{N_{\text{exp}}} \Delta t_i = \lim_{N_{\text{exp}}\to\infty} \left[\frac{1}{2^{N_{\text{exp}}}}\Delta t_{N_{\text{exp}}} + \frac{1}{2^{N_{\text{exp}}-1}}\Delta t_{N_{\text{exp}}} + ... + \frac{1}{2^0}\Delta t_{N_{\text{exp}}}\right] = \lim_{N_{\text{exp}}\to\infty} \sum_{j=0}^{N_{\text{exp}}} \left(\frac{1}{2}\right)^i \Delta t_{N_{\text{exp}}} = \frac{\Delta t_{N_{\text{exp}}}}{1 - \frac{1}{2}} = 2\Delta t_{N_{\text{exp}}}, \tag{2.1}$$

where the second last equality holds since we are facing a *converging geometric series*. This shows that if we change the exposure times by a power of 2, the maximum time it takes to capture the exposure sequence is bounded, and does not exceed twice the longest exposure time. In practice, changing the exposure time with the *uEye* cameras takes a non-neglectable amount of time, so that the above theoretical bound does not hold in our setup.

2.2 Spatial Stereoscopic HDR

In this mode, the left and the right camera are set to a fixed, distinct exposure time. We denote the right and left camera c_1 and c_2 respectively, and their corresponding exposure times Δt_{c_1} and Δt_{c_2}. Without loss of generality, we set $\Delta t_{c_1} \leq \Delta t_{c_2}$, i.e. the right camera will be the one with shorter exposure time. The data needs to be fetched from images that were taken with a horizontal baseline. This implies that some parts of the scene

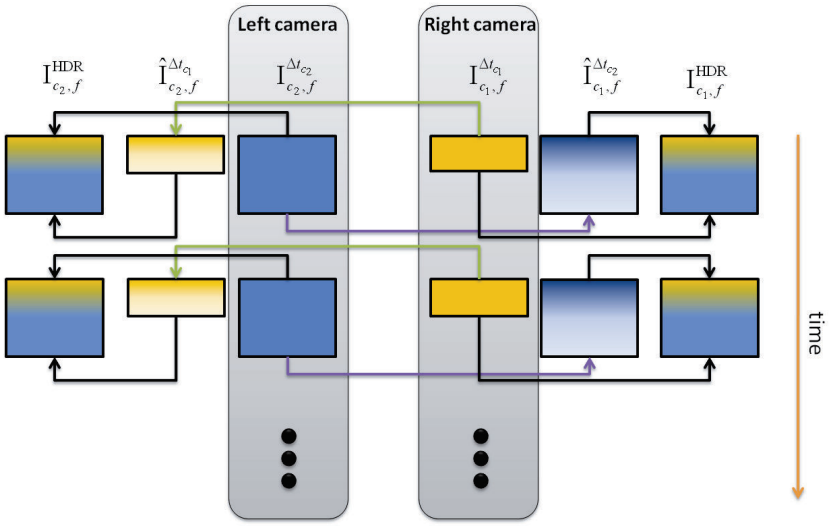


Figure 2.3: The Spatial stereoscopic HDR uses information from the left view to enhance the dynamic range of the right view and vice versa.

will be half-occluded, which complicates the correspondence problem compared to the temporal approach.
It is important that both cameras capture the same time instance. This requires that the right camera waits for the left camera until they are both ready to capture the next frame. The details of how this synchronization has been achieved are explained in Section 2.5.

2.3 Stereoscopic HDR Setup with Three Cameras

Even though time did not allow to implement and evaluate setups with three cameras, we still found it worth mentioning one of the setups using three cameras we had in mind, along with the potential benefits of adding a third camera.

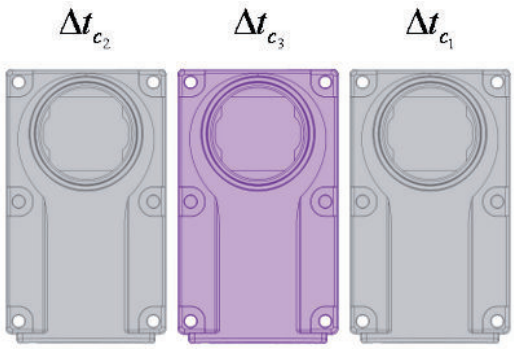


Figure 2.4: General setup using three cameras.

2.3.1 Three Cameras at the same Horizontal Baseline

This setup consists of three cameras on the same horizontal line. The exposure times would be set such that $\Delta t_{c_2} = \Delta t_{c_1} = \Delta t_{\text{ref}}$. This means that the left and the right camera would operate in traditional stereo setup. The additional third camera would be used as a helper camera in order to enhance the dynamic range. The basic idea is that disparity estimation would be done between the right and the left camera, which are set to the exposure time one would choose if just one exposure time was available. This would allow for a more reliable disparity estimation than if the two exposure times were different. The third camera would alternate between two exposure times Δt_1 and Δt_2, where $\Delta t_1 < \Delta t_{\text{ref}} < \Delta t_2$. Since the third camera is in the middle of the left and the right camera, disparities could be quite easily computed based on the disparites found between the left and the right camera. Figure 2.5 shows how the dynamic range could be enhanced by using this setup, in the example of enhancing the right view.

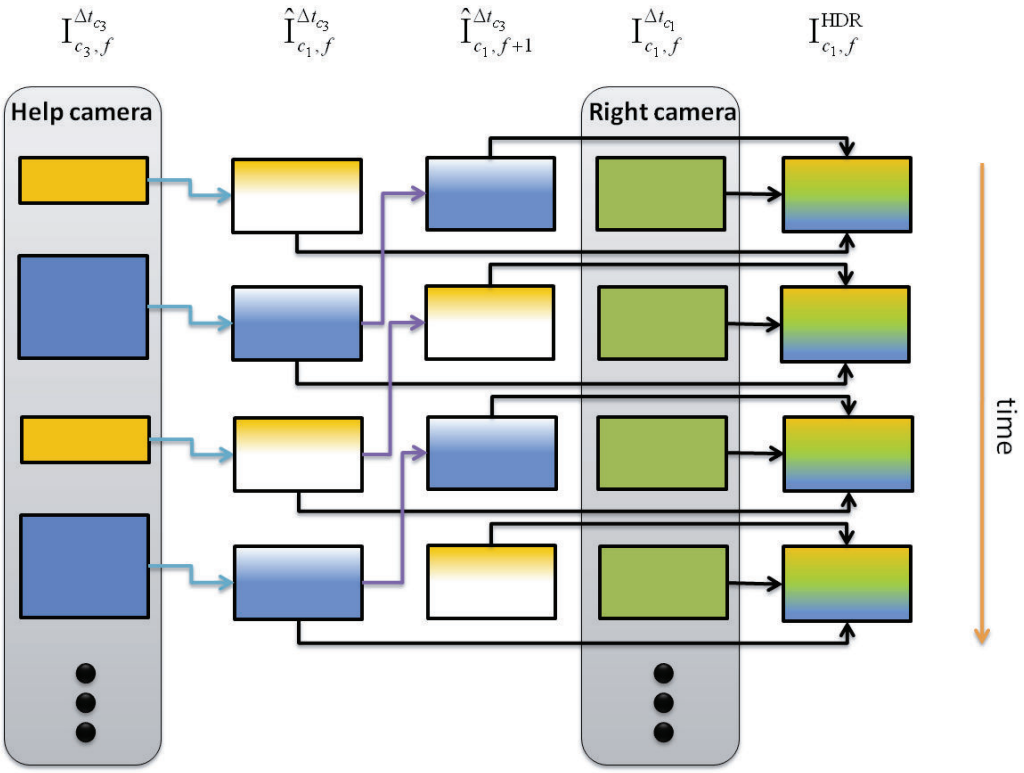


Figure 2.5: How the dynamic range is enhanced for the right camera. Arrows in cyan indicate disparity estimation and purple arrows indicate motion compensation.

First, the current frame f and the next frame $f+1$ would be warped into the reference of the right camera, which is indicated by the cyan arrows. The next frame then needs to be motion compensated, which is indicated by the purple arrows. The three aligned exposures can then be merged to an HDR radiance map.

2.4 Experimental Setup

2.4.1 Computer and Software

All experiments were carried out on a Core2Duo with 3 GHz, using Windows XP Professional with Service Pack 3. Development has been done in C++, making use of OpenCV 2.1 for image operations and the *uEye* API for handling the camera I/O. In addition, Matlab was used to create plots and for some parts of the project, especially for the evaluation in Chapter 4.

2.4.2 Cameras

We used *uEye* cameras from the German manufacturer IDS. Table 2.1 shows the specifications of the cameras used in the setup [13].

Model number	UI-2230ME-C
Interface	USB
Lens mount	C-Mount
Resolution	1024 x 768 (XGA)
Resolution depth	8 bit (12 Bit ADC)
Sensor size	1/3”
Maximum framerate	30 fps

Table 2.1: Specifications of the used cameras.

In the following, the basic functioning of the *uEye* cameras is explained. There exist two capturing modes, namely the *freerun* and the *trigger* mode. In *freerun* mode, the camera sensor captures one image after the other at a set frame rate. While the current image is being exposed, the previous is read out and transferred to the computer. This mode has to be used in order to achieve the maximum framerate of 30 fps. The downside of the *freerun* mode is that exposure time changes are guaranteed only for the frame after the next one. The *trigger* mode achieves lower maximum framerates than the *freerun* mode as capturing and data transfer are serialized, but exposure time changes are guaranteed for the next frame to be captured. Since the maximum framerate that can be achieved using the cameras is 30 fps, the maximum framerate using alternating exposure times in freerun mode would be 15 fps, as exposure time changes are not guaranteed for the subsequent frame. We therefore decided to use the *trigger* mode, where exposure time changes are immediately applied.

2.4.3 Lenses

The lenses used are from the manufacturer Computar [4]. Table 2.2 shows the relevant specifications.
It has to be noted that one of the two lenses seemed to have a little defect, resulting in the fact that captures from the left and the right camera were not perfectly aligned vertically. Section 3.2 explains how this has been corrected.

Model number	H0514-MP2
Focal length	5 mm
Aperture ratio	f/1.4-16
Focus range	0.1 m - 0.9 m

Table 2.2: Specifications of the lenses used.

2.4.4 Camera Rig

The camera rig was designed and constructed in a way that it allows an easy (un)mounting of the cameras. It consists of three main parts: A metal plate where the cameras are fixed, a Manfrotto 190XP aluminium tripod, and two metal bars that are used to fix the metal plate on the tripod. Figure 2.6 shows the blueprint of the metal plate, made in GIMP.

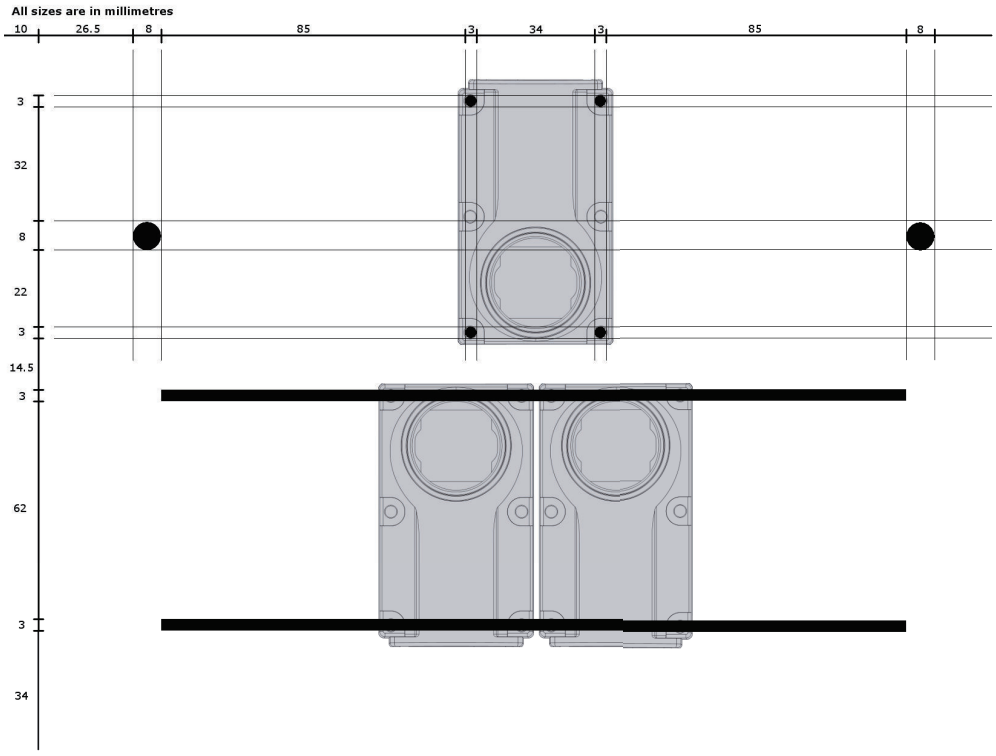
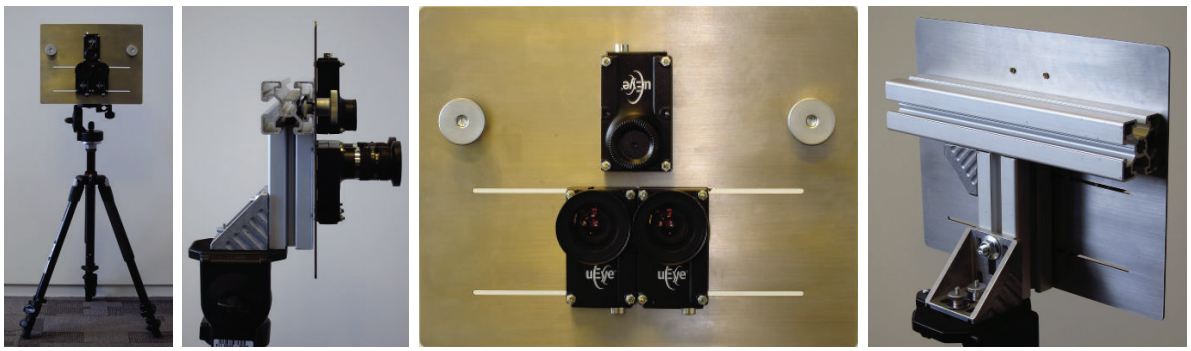


Figure 2.6: Blueprint of the possible positions of the cameras. While the position of the upper camera is fixed, the lower cameras can be moved to the left and to the right, allowing to increase/decrease the horizontal baseline. One can also easily add a third camera in the lower part.

The camera rig can be easily moved around and is fixed on the tripod such that the cameras can be freely rotated horizontally and vertically. The metal plate can also easily be removed from the tripod if a "freehand" mode is required. Figure 2.7 shows different views of the camera rig.
The cameras are fixed to the metal plate by four screws and can be easily unmounted and changed in position.
For all the reference captures used in this project, a horizontal baseline of 3.5 cm was used,



(a) Front view (b) Side view (c) Close-up on metal plate with cameras (d) Metal bars used to fix the metal plate on the tripod

Figure 2.7: Camera rig used in the experiments.

which is the minimal baseline possible for this setup. This was done since we experienced too much depth for the recorded scenes with the traditional baseline of 6.5 cm.

2.5 Synchronization of Cameras

When capturing frames, a significant amount of time is spent processing the captured frame and transferring it to the computer. Also, the cameras are not synchronized by default. For our setup, it is vital that they capture data at the same time, not sequentially, preferably at a high frame rate. There are several possible ways to achieve this. On the hardware side, one could use a hardware trigger which sends an electrical signal to the cameras at the exact same time, which then triggers them to take a capture. In order to make sure that sending the images to the computer does not present a bottleneck, one could add a frame grabber. Both these methods are quite costly which is why we decided to use a software approach, which is explained in more detail in the following.

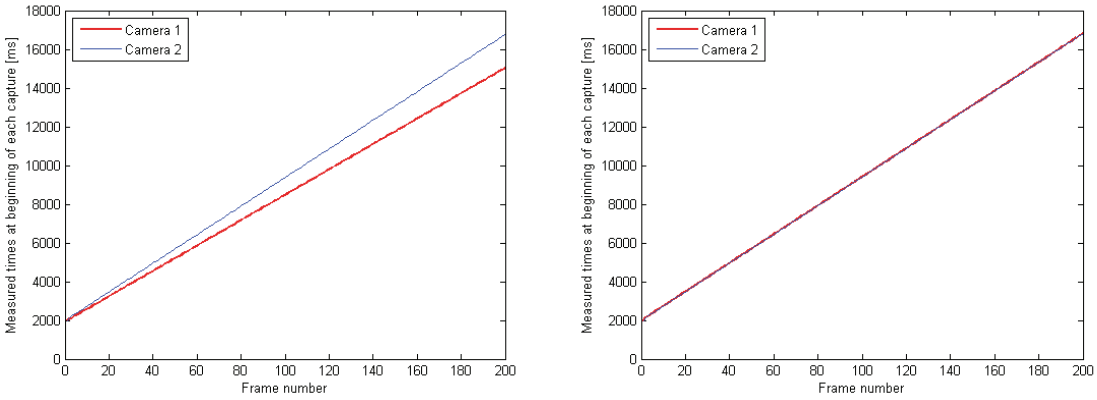
2.5.1 Multi-Threading to synchronize Captures

The CPU of a computer can only execute one task at a time. This implies that simply issuing the capture commands for the two cameras one after the other would result in captures that are not capturing the same moment in time in the left and the right camera. Several tasks can be executed in parallel by using multiple threads, which are then in turns processed by the CPU in a sequential manner. For our application, we decided to use one thread per camera, which are then in parallel capturing the same scene. In the following, the steps towards a synchronized capturing of the cameras are explained.

2.5.1.1 One Thread per Camera

Threads are by default executed independently. The first attempt to just start one thread per camera expectedly resulted in unsynchronized captures, which is very undesirable. Figure 2.8(a) shows the recorded times before each capture for the two cameras, where

the exposure time of the left camera was four times longer than the one of the right camera.



(a) Two unsynchronized threads.

(b) Two synchronized threads.

Figure 2.8: Difference between synchronized and unsynchronized threads. 200 frames were captured, with $\Delta t_1 = 6.71$ ms, $\Delta t_2 = 4 \cdot 6.71 = 26.8$ ms. One can clearly see how the capturing gets out of phase for the unsynchronized threads, which is very undesirable for this project. Note how the thread for camera one slows down when the cameras are synchronized.

One can clearly see how they get more and more out of synchronization. We therefore needed a way to signal to the other camera that it was ready for a new capture.

2.5.1.2 Synchronized Threads

In Figure 2.8(b) we can see how the camera with the shorter exposure time waits until the one with the longer exposure time also has finished recording. This synchronization is achieved by using two semaphores [23]. Figure 2.9 shows the flow diagram for the two threads.

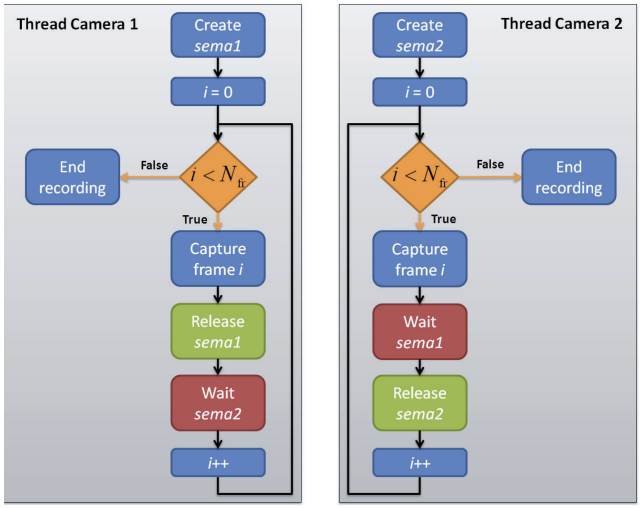


Figure 2.9: Flow diagram for the synchronization of the two cameras.

Each camera thread creates a semaphore, denoted as *sema1* and *sema2*, with a maximum count of 1, and initialized to 0. Then the first frame is captured for both cameras. Since the exposure time of camera 1 is shorter than the one of camera 2, the capturing time will be shorter. Camera 1 will then release *sema1*, which increases its count to 1. It then has to wait for *sema2* to be released. This is only done once camera 2 has finished capturing frame i. The thread of camera 2 will then wait for *sema1* to be released, which normally already is the case as camera 1 has already released *sema1*. It will then release *sema2*, at which point both threads will increase their frame counter and capture frame i+1.

2.6 Conclusion

In this chapter, the two main recording modes have been explained. Synchronization between the two cameras is crucial in order to record consistent left and right views. A software method using threads has been chosen in order to achieve this. Tests on actual recordings have shown that it works as desired. The spatial approach is able to record at a higher temporal sampling frequency than the temporal one, since in the spatial mode, the exposure time for one HDR image is limited by the long exposure time, whereas for the temporal mode, it is the sum of the exposure times. The fact that the exposure time changes take time also favors the spatial approach, as here the exposure time per camera is fixed. This allows to capture faster motion. One can easily compute how much higher the sampling frequency is for the spatial approach. Let us define the exposure ratio as $e = \frac{\Delta t_{c_2}}{\Delta t_{c_1}}$, and denote as $r = \frac{\Delta t_{\text{temp}}}{\Delta t_{\text{spat}}}$ the temporal sampling frequency ratio. Then,

$$r = \frac{\Delta t_{\text{temp}}}{\Delta t_{\text{spat}}} = \frac{\Delta t_{c_1} + \Delta t_{c_2}}{\Delta t_{c_2}} = \frac{\Delta t_{c_1} + e\Delta t_{c_1}}{e\Delta t_{c_1}} = \frac{\Delta t_{c_1}(1+e)}{\Delta t_{c_1}(e)} = \frac{1+e}{e}. \tag{2.2}$$

Note that this ratio gets smaller as e increases. In the limit, we arrive at:

$$\lim_{e\to\infty} \frac{\Delta t_{\text{temp}}}{\Delta t_{\text{spat}}} = \lim_{e\to\infty} \frac{1+e}{e} = 1. \tag{2.3}$$

While the limit above is theoretical, we can see in Table 2.3 how fast the temporal sampling frequency ratio approaches 1.

e	1	2	4	8	16
r	2	1.5	1.25	1.125	1.0625

Table 2.3: Table showing how the temporal sampling frequency ratio gets smaller with increasing exposure ratio.

3

Processing

Once the frames are captured, they need to be processed in order to create HDR radiance maps for the left and the right view. This includes the estimation of the camera response function, alignment of images, merging them to an HDR radiance map, and then tone mapping to adapt the content to the displayable gamut of the viewing device. All steps except the image alignment part (see Figure 3.1) are the same for the temporal and the spatial approach.

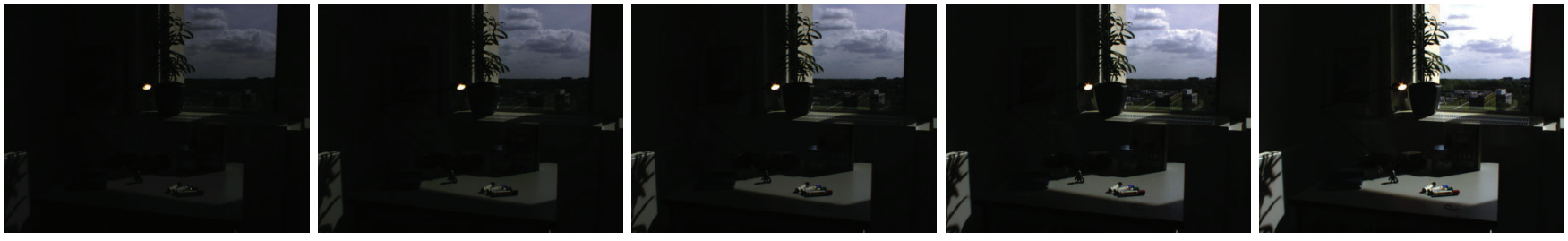
Figure 3.1: Parts of the processing block.

3.1 Camera Response Function (CRF) of an IDS Imaging UI-2230ME-C Camera

In order to be able to compute the radiance map out of an exposure series, the inverse camera response function needs to be known. Since we are capturing the raw data coming from a *charge-coupled device (CCD)* sensor which is inherently linear, we expect the camera response function to be linear. In order to confirm this assumption, we used the Matlab code provided by Debevec and Malik [5] to find g, the inverse of the camera response function. We took an exposure series of $N_{\text{exp}} = 15$ images, with exposure times increasing at $\frac{1}{2}$ stops, i.e.

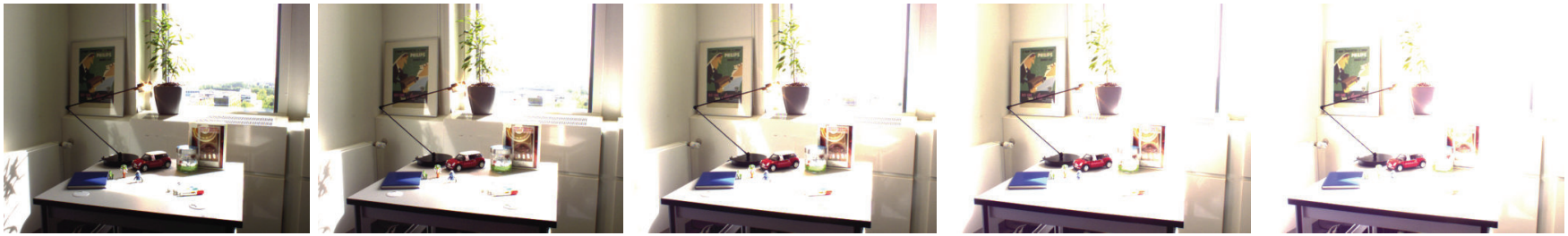
$$\Delta t_1 = a \tag{3.1a}$$

$$\Delta t_{i+1} = 2^{\frac{i}{2}} \Delta t_1, \forall i \in [1, N_{\text{exp}} - 1], \tag{3.1b}$$



(a) $\Delta t_1 = 0.989$ ms (b) $\Delta t_2 = 1.433$ ms (c) $\Delta t_3 = 2.005$ ms (d) $\Delta t_4 = 2.830$ ms (e) $\Delta t_5 = 3.973$ ms

(f) $\Delta t_6 = 5.688$ ms (g) $\Delta t_7 = 7.974$ ms (h) $\Delta t_8 = 11.339$ ms (i) $\Delta t_9 = 15.975$ ms (j) $\Delta t_{10} = 22.642$ ms



(k) $\Delta t_{11} = 31.977$ ms (l) $\Delta t_{12} = 45.248$ ms (m) $\Delta t_{13} = 63.981$ ms (n) $\Delta t_{14} = 90.524$ ms (o) $\Delta t_{15} = 127.989$ ms

Figure 3.2: Raw reference exposure series (not gamma-corrected) captured to estimate the camera response function, as well as for evaluation purposes. Aperture f/8.0, $a = 0.989$ ms.

where a is a constant depending on the lighting conditions of the captured scene, chosen such that details in the bright areas of the scene are captured. We refer to this scene as *ground truth*. Figure 3.2 shows the raw captured images of the right camera along with the true exposure times.

3.1.1 How the Pixels to determine the CRF were selected

Instead of manually selecting pixels as proposed by Debevec and Malik [5], we took up Reinhard *et al.*'s [33] idea and looked at the cumulative histograms of the images, and selected pixel values based on percentiles. Figure 3.3 shows the cumulative histogram for the green color channel of every second exposure of the ground truth image set for the right camera. Similar results are obtained for the other color channels, as well as for the left camera.

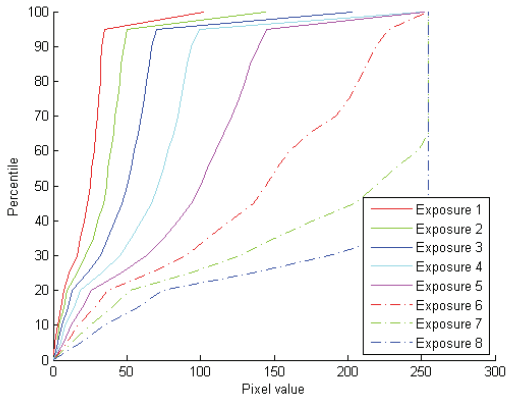


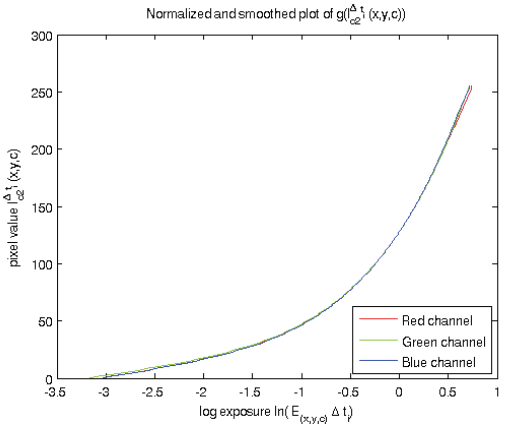
Figure 3.3: Cumulative histogram for the green channel of every second exposure of the right camera.

The pixel values were then selected in all the 15 exposures according to the percentiles 0% to 100%, with steps of 5%. Not only is this approach more robust to outliers, but it also has the great advantage that in case there is a slight movement of the camera, we still get the same camera response function.

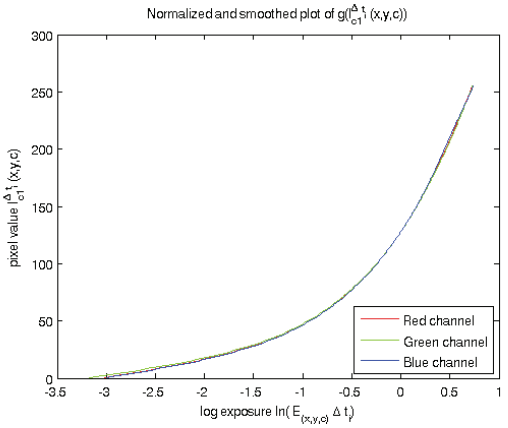
3.1.2 Linearity of the CRF

In order to estimate the radiance map, only the inverse CRF is needed. For reasons which will be explained in Section 3.3, we also need to perform the opposite conversion, i.e. we need the CRF. While Debevec and Malik's [5] method provides a mapping from pixel values to radiance, the inverse is not the case. In other words, the CRF needs to be approximated. Mangiat and Gibson [19] do so by fitting an exponential of the form $Ae^{BE_i} + C$ to g^{-1}.
In our case, the inverse CRF is almost perfectly linear for all three color channels, as can be seen in Figure 3.4. Together with the fact that CCD sensors are inherently linear, the only question left was to find out what happened by the 12 bit ADC to 8 bit conversion done by the camera. As can be read in the manual of the *uEye* camera [12], only the 8 most significant bits are kept, keeping the response linear. We therefore assume linearity

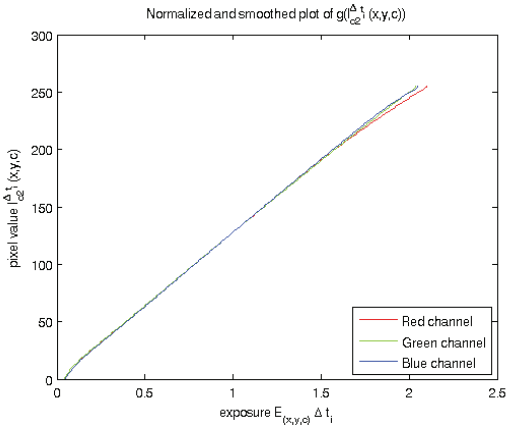
of the camera response function, which makes its inversion much more practical (see Section 3.3).



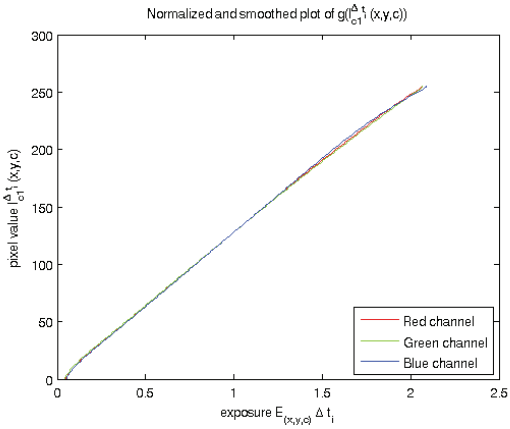
(a) Left camera: Logarithmic scale



(b) Right camera: Logarithmic scale



(c) Left camera: Linear scale



(d) Right camera: Linear scale

Figure 3.4: Estimated camera response function of the camera setup used in this thesis.

3.2 Pre-Processing Images

Before we start with the actual processing of the images, a few pre-processing steps have been applied to the images in order to give nice results on a stereoscopic display. In fact, the cameras had a slight vertical misalignment, which was corrected by cutting the right image at the top and the left image at the bottom. Also, in order for all objects in the scene to be behind the screen in order to avoid border violations, both images were shifted by the maximum disparity present in the scene, which for the ground truth set was 50 pixels.

3.3 Simulated Re-exposure of Pixel Values

In the context of stereoscopic HDR, it is important to be able to predict whether a pixel captured at a given exposure time is going to be clipped in an image of the same scene, captured with a different exposure time. We further call this simulated re-exposure of the pixel value simply re-exposure. Let $\mathrm{I}_{c_j}^{\Delta t_i}(x, y, k)$ be the value of pixel at coordinate (x, y) of color channel k captured with an exposure time Δt_i. Assume we want to re-expose pixel (x, y, k) captured at exposure time Δt_{old} to simulate an exposure time Δt_{new}. In the following, it is explained how this can be done in the case of both linear and non-linear *camera response function (CRF)*.

3.3.1 Linear Camera Response Function

In the case the camera response function is linear, re-exposing pixel values is quite easy. In fact, one can compute the new pixel value using the following formula:

$$\mathrm{I}_{c_j}^{\Delta t_{\text{new}}}(x, y, k) = \min(\mathrm{I}_{c_j}^{\Delta t_{\text{old}}}(x, y, k))\frac{\Delta t_{\text{new}}}{\Delta t_{\text{old}}}, 255.0). \tag{3.2}$$

This is the formula we used in this project to re-expose pixel values since the camera response function has shown a highly linear behaviour. Note that a re-exposed value can not exceed 255.0, as we know that this is the maximum value in the other image.

3.3.2 Non-linear Camera Response Function

If the camera response function is not linear, its inversion is not trivial. In order to correctly re-expose a pixel value, one needs to first apply the inverse camera response function, then apply the exposure ratio, and then apply the camera response function to get back to the pixel domain. Equation 3.3 shows how this can be done.

$$\mathrm{I}_{c_j}^{\Delta t_{\text{new}}}(x, y, k) = \min(f(f^{-1}(\mathrm{I}_{c_j}^{\Delta t_{\text{old}}}(x, y, k)) + \ln(\frac{\Delta t_{\text{new}}}{\Delta t_{\text{old}}})), 255.0). \tag{3.3}$$

Figure 3.5 visualizes the process of re-exposing a pixel value. Depending on the ratio $\frac{\Delta t_{\text{new}}}{\Delta t_{\text{old}}}$, the pixel value $\mathrm{I}_{c_j}^{\Delta t_{\text{old}}}(x, y, k)$ gets re-exposed accordingly.
In order to create a correct HDR image, the images that are merged together to the HDR radiance map need to be correctly aligned, as misaligned images lead to undesired ghosting artifacts. The processing to align images differs greatly between the temporal and the spatial stereoscopic HDR approach. In the following two sections, we build up a pipeline that allows to create HDR left and right views for both methods.

3.4 Image Alignment for Spatial Stereoscopic HDR: Disparity Estimation

Since the two images used to create the HDR radiance map are taken by two different, synchronized cameras, they capture the same instant in time (see Section 2.2). Under the assumption that the long exposure time is short enough that there is no motion

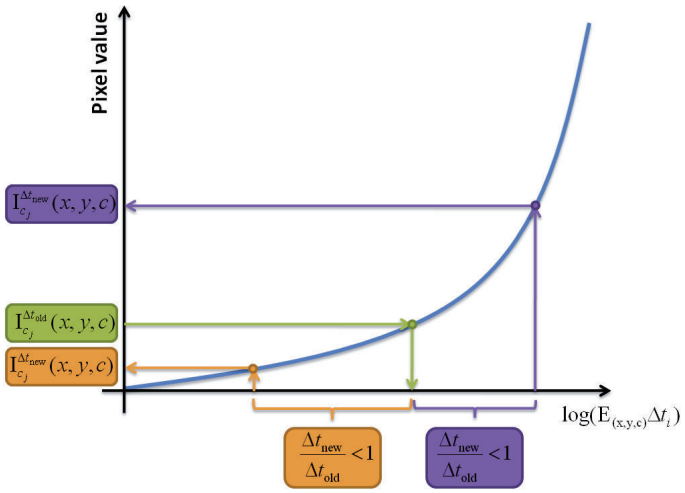


Figure 3.5: Visualization of the re-exposing process of a pixel.

blur, motion compensation is not needed. But the fact that two images are taken from two different points of view implies that only objects at infinity are correctly aligned. Disparity estimation is therefore needed in order to align the images, even if there is no motion in the captured scene.
We start with a traditional block-based disparity estimator, and then gradually add refinements in order to improve the performance. It is worth mentioning that in the case of different exposure times for the two views, the disparity maps will not be the same due to clipping. In the following, the different steps for the enhancement of the long exposure are explained, as the motivations for changing the algorithm are more visible. Figure 3.6 shows the two images used to create the results in this section.

(a) Left view with an exposure time of Δt_8. (b) Right view with an exposure time of Δt_4.

Figure 3.6: Left and right view of the reference scene used in the following discussion. Note how the sky and parts of the table and the wall are clipped in the long exposure.

3.4.1 Traditional Block-based Disparity Estimation using SAD

The first approach is to take a traditional block-based disparity estimator using *sum of absolute differences (SAD)* as measure of similarity. The current image is partitioned

into N_{reg} disjoint blocks. We denote block s of an image taken by camera j as $S_{c_j,s}$. We further denote $C_s^{(c_1,c_2)}(\Delta i, \Delta j)$ the SAD for $S_{c_1,s}$ with a block shifted by a displacement vector $(\Delta i, \Delta j)$ in an image taken by camera 2. Equation 3.4 shows how the SAD is computed.

$$C_s^{(c_1,c_2)}(\Delta i, \Delta j) = \sum_{i \in S_{c_1,s}} \sum_{j \in S_{c_1,s}} \sum_{k \in \{R,G,B\}} |\mathrm{I}_{c_1}^{\Delta t_1}(i,j,k) - \mathrm{I}_{c_2}^{\Delta t_2}(i+\Delta i, j+\Delta j, k)|. \quad (3.4)$$

We are now interested in finding the best matching blocks in a horizontal and vertical search range, denoted as Δj and Δi respectively. As explained in Section 3.2, the images have been shifted to "remove" vertical disparities and to bring everything at the back of the screen. This implies that vertical search range can be very small, and the horizontal search range is bounded by 0 on one side. In our setup, we used $\Delta i \in [-2, 2]$, $\Delta j \in [-50, 0]$ for right to left matching, and $\Delta j \in [0, 50]$ for left to right matching. The optimization is based on a *winner takes it all (WTA)* strategy, i.e. the disparity with the smallest SAD is registered as the disparity of the block [37]. The two-element disparity vector for region s from camera c_1 to camera c_2 is denoted as $\vec{v}_s^{(c_1,c_2)}$. This vector, which indicates the location from where the information in the other view is to be fetched, is computed as:

$$\vec{v}_s^{(c_1,c_2)} = (\Delta i^{\min}, \Delta j^{\min}) = \underset{\Delta i, \Delta j}{\operatorname{argmin}} \, [C_s^{(c_1,c_2)}(v,h)]. \quad (3.5)$$

Figure 3.7 shows the block-diagram of a traditional block-based disparity estimator. First, $\mathrm{I}_{c_2}^{\Delta t_2}$ is partitioned into blocks in the segment step. For each block, the disparity vector corresponding to the min SAD is computed. In the last step, the data is being fetched from $\mathrm{I}_{c_1}^{\Delta t_1}$.

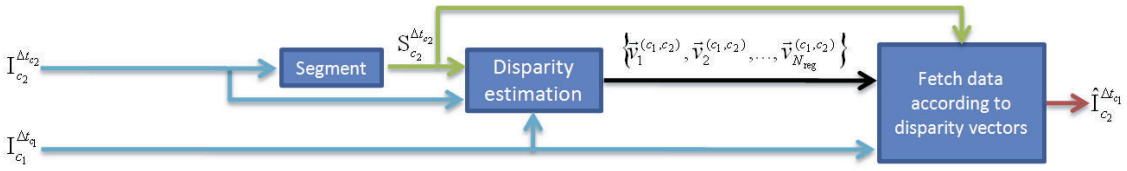
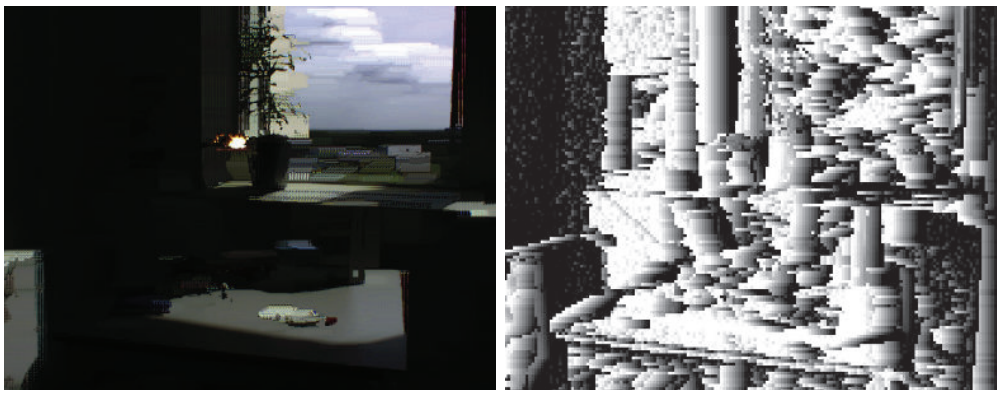


Figure 3.7: Traditional block-based disparity estimation used to enhance $\mathrm{I}_{c_2}^{\Delta t_2}$.

Figure 3.8 shows $\hat{\mathrm{I}}_{c_2}^{\Delta t_1}$, the fetched information from the short exposure. One can see repetitive patterns in the fetched image.

3.4.2 Re-exposure of Images to simulate same Exposure Time

As we have seen in the previous part, the fact that the images have different exposures leads to very bad matches as the intensity values are different. One way to fix this problem this problem would be to use another metric such as *normalized cross correlation (NCC)*, which has been shown to be intensity invariant (Troccoli *et al.* [40]). This means that it matches well whenever the texture matches. On top of that, the NCC is much more computationally expensive, which is why we choose to stick with the SAD, and to do a different modification to account for the intensity differences in the two images due to



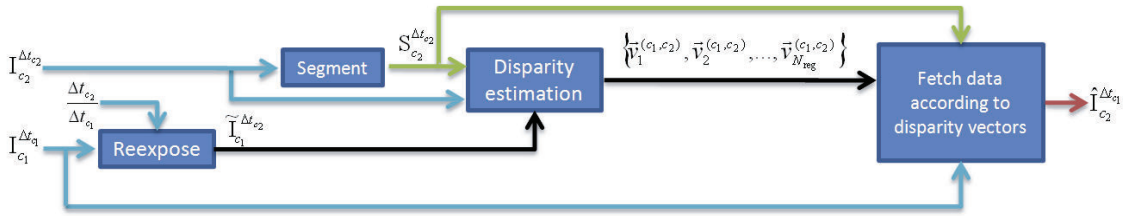
(a) Fetched data (b) Horizontal disparities

Figure 3.8: Fetched data and the horizontal disparities obtained with the traditional block-based disparity estimator with a blocksize of 6x6 pixel, $\Delta t_{c_1} = \Delta t_4$ and $\Delta t_{c_2} = \Delta t_8$.

the different exposure times. We re-expose the shorter exposure as described in Section 3.3, in order to have the same simulated exposure time as the long exposure. In other words, the computation of the matching cost shown in Equation 3.4 is changed to:

$$C_s^{(c_1,c_2)}(\Delta i, \Delta j) = \sum_{i \in S_{c_1,s}} \sum_{j \in S_{c_1,s}} \sum_{k \in \{R,G,B\}} |\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}(i,j,k) - \mathrm{I}_{c_2}^{\Delta t_2}(i+\Delta i, j+\Delta j, k)|, \qquad (3.6)$$

where $\mathrm{I}_{c_1}^{\Delta t_1}(i,j,k)$ has been re-exposed to $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}(i,j,k)$. This results in the modified fetching procedure shown in Figure 3.9.



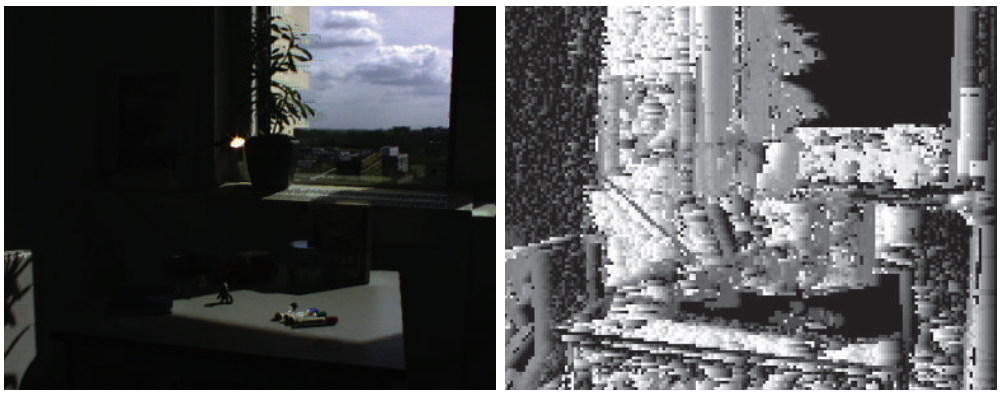
(a) Block-diagram

Figure 3.9: Block-based disparity estimation where the short exposure image has been re-exposed. Blocksize 6x6 pixel, $\Delta t_{c_1} = \Delta t_4$ and $\Delta t_{c_2} = \Delta t_8$.

The short exposure $\mathrm{I}_{c_1}^{\Delta t_1}$ is re-exposed to obtain the simulated long exposure $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}$, which should match the intensity values of $\mathrm{I}_{c_2}^{\Delta t_2}$ modulo quantization and noise. The disparity estimation is now done between $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}$ and $\mathrm{I}_{c_2}^{\Delta t_2}$, which leads to much better results as can be seen in Figure 3.10.

3.4.3 Region Classification

Thus far, we have completely ignored the fact that some regions are clipped in the dark and/or the bright part. Those regions are in fact the most problematic ones, since they contain valuable information only in one of the two views. In other words, where there is



(a) Fetched data (b) Horizontal disparities

Figure 3.10: Fetched data and the horizontal disparities obtained with the block-based disparity estimator and re-exposure of the short exposure, with a blocksize of 6x6 pixel, $\Delta t_{c_1} = \Delta t_4$ and $\Delta t_{c_2} = \Delta t_8$.

likely to be texture in one image, the same object might be clipped in the other view and hence contain no texture at all. For this reason, any matching algorithm will be bound to fail in clipped regions. Figure 3.11 visualizes the effect of half-clipping. If we segment exposure Δt_8 and want to match it with exposure Δt_4, for all pixels that are in the blue colored rectangle, there will be no data present in the target image. Vice versa, all pixels in the orange rectangle will be clipped in the segmented image.

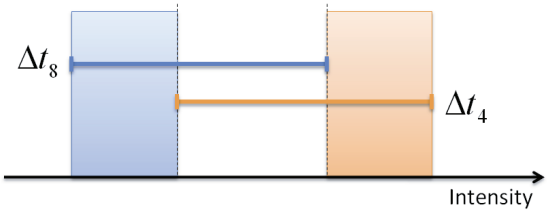


Figure 3.11: The exposures taken at Δt_4 and Δt_8 capture different parts of the luminance range, resulting in pixels that are clipped in one view but not the other.

The next refinement consists in classifying the set of regions Ω into three disjoint subsets Ω_i for $i \in \{1, 2, 3\}$, by assigning to each region a label L_j $\forall j \in \{1, ..., N_{\text{reg}}\}$. We define a region as clipped if more than 50% of the pixels belonging to that region are clipped in all three color channels. Figure 3.12 shows the pipeline for fetching information from the short exposure in order to enhance the long exposure.
According to their label, the regions are subsequently treated differently. Ω_1 consists of all regions that are predicted to be clipped in the other view. In the case of enhancing the long exposure, these are all regions where $\tilde{\mathrm{I}}_{c_2}^{\Delta t_1} = 0$. For the other case, the criterion is $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2} = 255$. For these regions, we know that there will be no useful data and hence we are better off not fetching information from the other view, but instead keeping the one of the short exposure. For this, we re-expose the image data of the current image in order to match the intensity of the other exposure. This will lead to clipping most of the actual region, which will result in disregarding it in the HDR merging step presented in Section 3.6 due to the hat weighting function. This effectively only keeps the current,

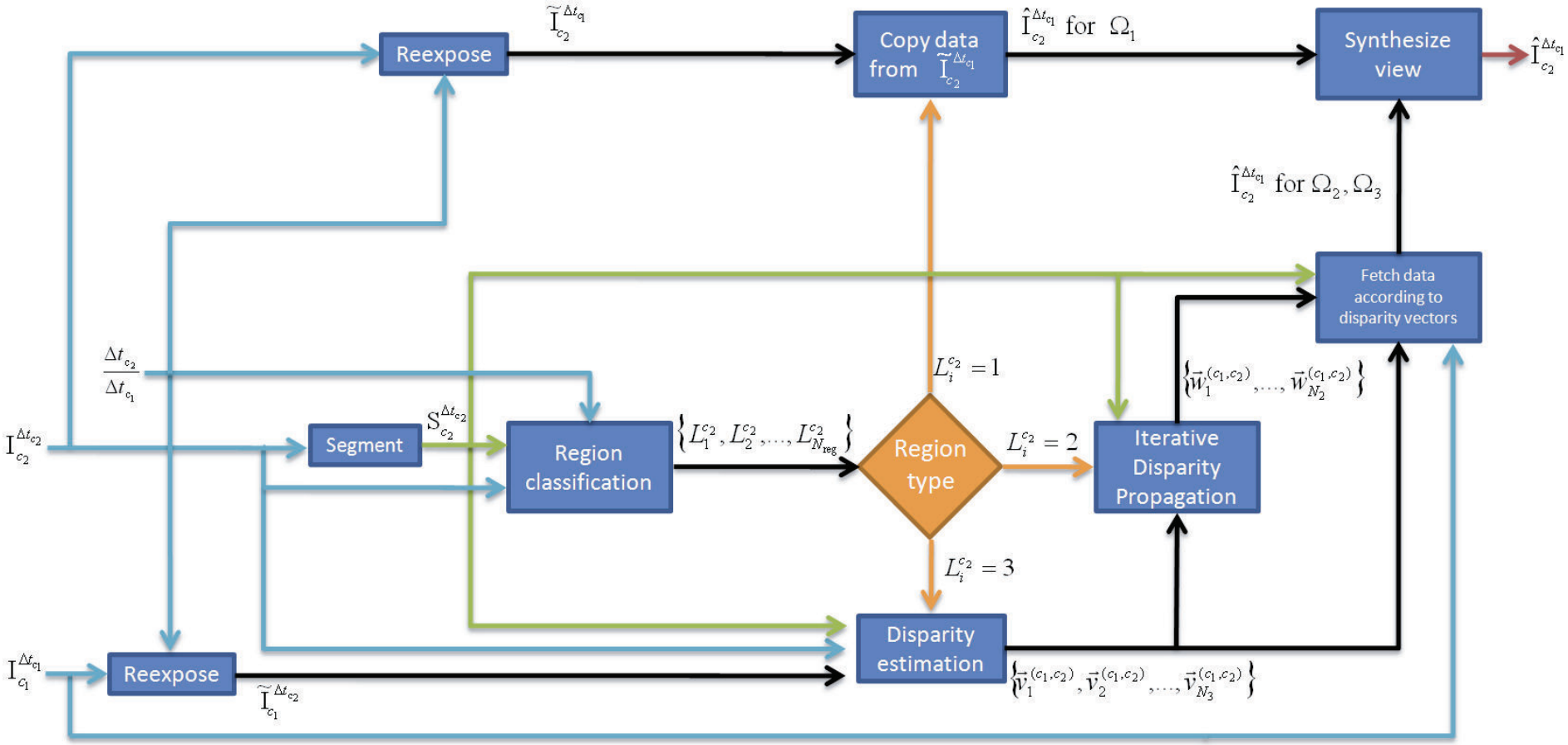
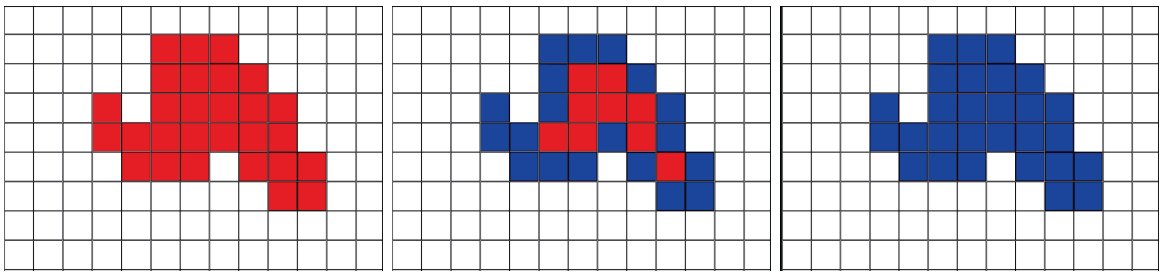


Figure 3.12: Block-based disparity estimation where regions are classified into three classes, which are subsequently treated differently. In this figure $\mathrm{I}_{c_2}^{\Delta t_2}$ is enhanced.

unclipped exposure.
Ω_2 are the regions that are clipped in the current view. This translates to $\mathrm{I}_{c_2}^{\Delta t_2} = 255$ and $\mathrm{I}_{c_1}^{\Delta t_1} = 0$ for the long and the short exposure respectively. For these regions, we have no useful information in the current view, which makes them the most difficult to handle. In fact, if they are not treated with care, the left and right view will not be consistent at all, as in one view the information will not be clipped, but in the other it will be.
Last but not least, Ω_3 are the regions that are neither clipped in the current view, nor predicted to be clipped in the other view. For these, disparity estimation is performed as in the previous examples, resulting in the set of disparity vectors $V = \{\vec{v}_1^{(c_1,c_2)}, \vec{v}_2^{(c_1,c_2)}, ..., \vec{v}_{N_3}^{(c_1,c_2)}\}$, where N_3 is the number of regions labelled as $L_j = 3$. All regions belonging to Ω_3 are marked as having a *valid disparity*. Together with the segmentation grid, V is then used to determine disparity vectors for regions belonging to Ω_2 using *iterative disparity propagation (IDP)*, which is explained in the following section. The output of the IDP is a set of estimated disparity vectors $W = \{\vec{w}_1^{(c_1,c_2)}, \vec{w}_2^{(c_1,c_2)}, ..., \vec{w}_{N_2}^{(c_1,c_2)}\}$, where N_2 is the number of regions belonging to Ω_2. For later references, we call U the union of V and W. In the next step, regions belonging to Ω_2 and Ω_3 use their assigned disparities in order to fetch the corresponding data from the other view. $\hat{\mathrm{I}}_{c_2}^{\Delta t_1}$ is completed by copying the missing regions directly from $\tilde{\mathrm{I}}_{c_2}^{\Delta t_1}$.

3.4.3.1 Iterative Disparity Propagation

This technique tries to unclip clipped regions by filling in the correct information from the unclipped image. Our approach is based on the assumption that the neighbors of clipped regions are the most likely to have a similar disparity. Areas that are clipped are being iteratively filled from the outside to the inside by taking the average of the disparity value of the adjacent regions that have a *valid disparity*, as shown in Figure 3.13.



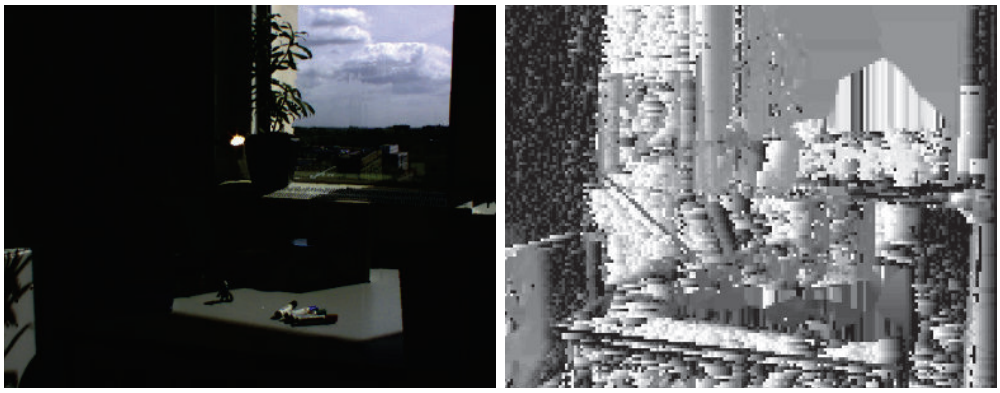
(a) Initial clipped region (b) After first iteration (c) After second iteration

Figure 3.13: How clipped regions are unclipped using *iterative disparity propagation* (IDP). White regions are regions with a valid disparity, red regions are the ones that are marked as being clipped. The blue regions are the ones that have been assigned disparities based on their neighbors.

At each iteration, all regions in Ω_2 that are adjacent to regions marked to have a *valid disparity* are assigned a disparity based on the average of all valid disparities adjacent to this region (see Equation 3.7).

$$\vec{w}_n^{(c_j,c_k)} = \frac{1}{|\Gamma_n|} \sum_{i \in \Gamma_n} \vec{u}_i^{(c_j,c_k)}, \tag{3.7}$$

where Γ_n contains the indexes in U of all regions adjacent to region n. After each iteration, all unclipped regions are marked to have *valid disparity*, such that clipped regions can shrink. This reduces the amount of regions belonging to Ω_2 in an iterative way until $\Omega_2 = \emptyset$. Due to its nature, we call this process of filling in the clipped holes IDP. Figure 3.14 shows the resulting picture after applying IDP.



(a) Fetched data (b) Horizontal disparities

Figure 3.14: Resulting fetched image and horizontal disparities after applying the IDP. Note how the clipped regions have similar disparities to their neighbors.

Note that only the clipped regions are being touched which before had a completely wrong disparity. In order to show the improvements, Figure 3.15 shows how the clipped regions improve when IDP is applied.

(a) Without IDP (b) With IDP

Figure 3.15: Comparison of the fetched short exposure with and without IDP. The red circles show parts of the image where the results improved with IDP.

For the IDP to work correctly, the regions with valid disparities adjacent to clipped regions need to be at the same depth in the scene. While this assumption mostly holds true for specularities on objects, it is not the case in general. For the scene at hand, the clipped sky is put at a wrong disparity since the frame of the window propagates wrong

disparities. Also, there is an even more fundamental problem that arises no matter how good the unclipping algorithm is, which is the fact that some clipped information is not present in the other view (half-occlusion). This is visualized in Figure 3.16, where the problematic part is highlighted in red.

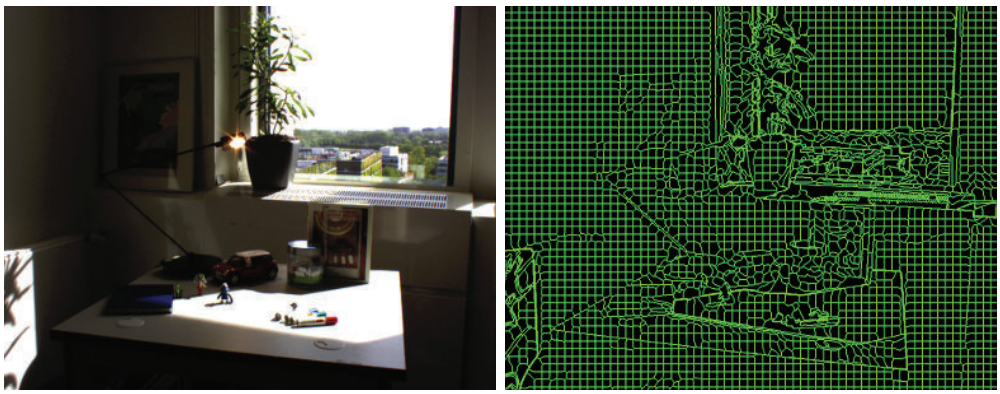
(a) Left view (b) Right view

Figure 3.16: Visualization of the fact that some clipped information is not present in the other view.

We refer to these regions as half-clipped, half-occluded. While already the detection of such regions is not trivial, it is an open problem to fill in the data in order to get a consistent stereo pair. One can see that in the sky region of the image in Figure 3.15(b), there are some visible block artifacts. The next section proposes a way to partially remove such artifacts by creating more meaningful regions than blocks.

3.4.4 Image Segmentation

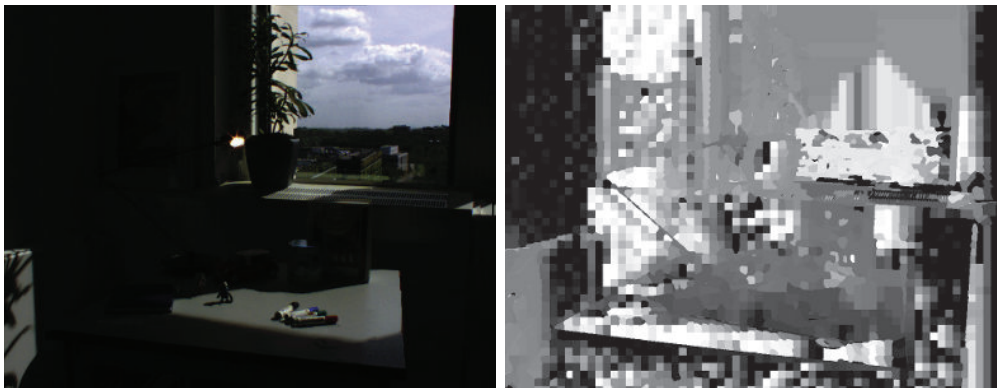
Block-based disparity estimation is known to lead to visible artifacts such as blockiness and foreground fattening. In the examples above, the block size is very small, so the block artifacts are hardly visible. The problem of using a small block size is that the results are less robust. One way to improve the results and (partially) remove the block artifacts and foreground fattening is to segment the image into more meaningful regions than blocks. This modification only changes the block "segment" of the block diagram in Figure 3.12, the rest is left untouched. An iterative segmentation algorithm was used [42]. This algorithm adapts the original region fitting idea that was proposed by Oliver and Quegan [27]. The newer method in [42] effectively deals with thin and long regions that occur on out-of-focus edges and solves this problem by introducing a majority filter step between two consecutive region fitting steps. The algorithm in [42] implements the iterative update using the global energy minimization as described by Duda *et al.* [7]. The result is an oversegmentation of the image that preserves the edges that are present, as can be seen on the example in Figure 3.17.
Note that in the case where the short exposure is to be enhanced, the segmentation is done on the re-exposed image $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}$. The reason for this is that this way, regions that will be clipped in the long exposure will also be clipped in $\tilde{\mathrm{I}}_{c_1}^{\Delta t_2}$, and hence there will not be any edges in these regions which would lead to segments.



(a) Original image (b) Segmentation

Figure 3.17: Result of segmenting, starting with a square grid of size 15x15.

Figure 3.18 shows the results of applying an oversegmentation to the image, resulting in more meaningful regions that respect object boundaries.



(a) Fetched data (b) Horizontal disparities

Figure 3.18: Resulting fetched image and horizontal disparities using image oversegmentation. Note how pixels belonging to the same object get assigned similar disparities. Initial segment size 15x15.

One can see that in regions with no texture, the disparities are likely to be wrong. This could be improved by using a more advanced disparity estimator.

3.5 Image Alignment for Temporal Stereoscopic HDR: Motion Estimation

By the nature of this setup, no image alignment is needed for still scenes. In case of motion, we need to apply motion estimation to find the motion field between the captures that will be merged together to an HDR radiance map. In order to develop and test the motion estimation algorithm, a scene of 200 frames has been recorded, from now on referred to as *car scene*. Figure 3.19 shows consecutive frames of the right camera operating in temporal HDR mode.

Figure 3.19: Consecutive frames of the captured *car scene* using the temporal mode, where $\Delta t_1 = 4.16$ ms and $\Delta t_2 = 33.50$ ms, resulting in an exposure ratio of $e = 8.05$.

The camera was fixed, and there is a car driving down a slope which was created using two metal bars. They served as rails, which in turns introduced friction so that the car had to be pulled using a thin thread. This led to a non-accelerated, jumpy movement of the car. The problem of motion estimation between differently exposed images has been addressed by several authors [17] [19]. It was not in the scope of their work to apply it to two cameras at the same time. In the interest of time, we implemented a basic motion estimation algorithm, assuming that there are moving objects in the scene, but that the camera itself is not moving.
Using the insights gained in Section 3.4, we directly implemented the re-exposure and classification parts, and used region segmentation instead of simple blocks. Figure 3.21 shows how the frames are processed, for enhancing a long exposure. Again, the general workflow for the enhancement of a short exposure is the same. The regions are classified into the same three subsets as for the disparity estimation algorithm explained in Section 3.4.3. Recall that Ω_1 consists of all regions that are predicted to be clipped in the other view, Ω_2 are the regions that are clipped in the current view, and Ω_3 are the regions with valid disparity vectors. The main difference between the temporal and the spatial approach is how the regions belonging to Ω_2 are treated. While for the spatial approach, *iterative disparity propagation* was used in order to estimate the disparity vectors for clipped regions, we simply copy the information from the next frame in the temporal approach, under the assumptoin that the clipped part does not belong to a moving object. Figure 3.20 shows the tone mapped result when there is motion in the scene.

(a) No motion compensation. (b) With motion compensation.

Figure 3.20: Moving objects in a scene need to be aligned.

One can clearly see the need for motion compensation, as there are quite severe artifacts around the car due to the fact that the two exposures are not aligned. The proposed

motion estimation algorithm is able to compensate the motion of the car, and align the exposures.

3.6 Merging LDR Images to HDR

In the next step, the aligned LDR images are merged together to an HDR radiance map. Debevec and Malik's [5] method has been implemented for these purposes. Equation 3.8 shows how the HDR radiance map is computed in case of a linear CRF.

$$\mathrm{I}_{c_j}^{HDR}(x,y,k) = \frac{\sum_{i=1}^{N} \frac{\Delta t_{\mathrm{ref}}}{\Delta t_i} \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k) w(\mathrm{I}_{c_j}^{\Delta t_i}(x,y,k))}{\sum_{i=1}^{N} w(\mathrm{I}_{c_j}^{\Delta t_i}(x,y,k))}, \tag{3.8}$$

where Δt_{ref} is the reference exposure time (set to Δt_8 in our case) and $w()$ is the following hat weighting function:

$$w(\mathrm{I}_{c_j}^{\Delta t_i}(x,y,k)) = \begin{cases} \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k) & \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k) \leq 128 \\ 255 - \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k) & \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k) > 128 \end{cases} = \min(\mathrm{I}_{c_j}^{\Delta t_i}(x,y,k), 255 - \mathrm{I}_{c_j}^{\Delta t_i}(x,y,k)). \tag{3.9}$$

3.7 Tone Mapping of the HDR Radiance Map for Viewing Purposes

The radiance map computed in the previous step consists of floating point values that go beyond the maximum number that can be represented with nowadays most used encoding size of 8 bit, which is 255. In order to be properly displayed on a device with a lower dynamic range, such as a standard display, the values of the HDR image need to be mapped down to the display device. The technique to map down the HDR image to be displayed on a lower dynamic range device is called *HDR tone mapping*. Tone mapping depends on the application and intent, and several methods have been proposed. Since tone mapping is not the main scope of this project, we will not go into detail of that topic. The interested reader is referred to [3], which gives a comprehensive comparison and evaluation of different tone mapping techniques in terms of perception, and [6] which is a bit older but gives a nice overview over tone mapping techniques. There exist two main types of tone mapping algorithms, namely *global* and *local*. As the name suggests, global tone mapping applies a global mapping function to the whole image, i.e. it is spatially invariant. The problem with global tone mapping methods is that they overcompress the tonal range, resulting in loss of detail visibility and contrast. For this reason, we chose to use a local tone mapping method, which will be briefly explained in the following section.

3.7.1 Local Tone Mapping simulating the Retinal Model

We chose to use the local tone mapping algorithm described in Tamburrino *et al.* [38], which is inspired by the non-linear processing taking place in the retina on the cone

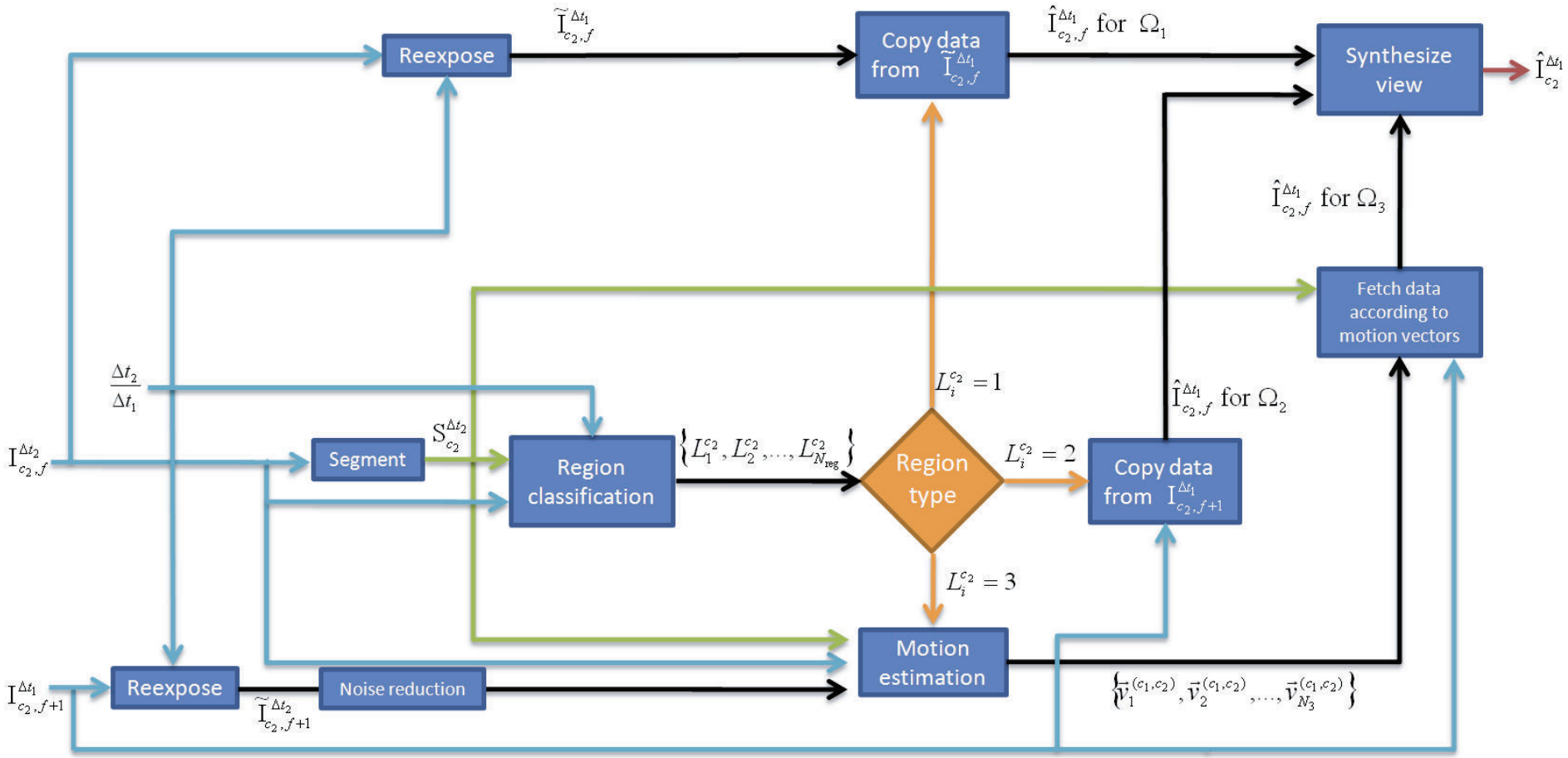


Figure 3.21: Region-based motion estimation where regions are classified into three classes, which are subsequently treated differently. In this figure $\mathrm{I}_{c_2}^{\Delta t_2}$ is enhanced.

mosaic, as described in Meylan *et al.* [22]. This algorithm uses a modified version of the *Naka-Rushton* function [26], in which the adaptation factor is pixel-dependent. This function is used to model the processing of both the *outer plexiform layer (OPL)* and the *inner plexiform layer (IPL)* of the retina. Equations 3.10 and 3.11 show the functions used to model the OPL and IPL processing, respectively, of the HVS:

$$\mathrm{I}_{c_j}^{bip}(x,y,k) = \underset{x,y}{\operatorname{argmax}}[\bar{\mathrm{I}}_{c_j}^{HDR}] + H(x,y))\frac{\mathrm{I}_{c_j}^{HDR}(x,y,k)}{\mathrm{I}_{c_j}^{HDR}(x,y,k) + H(x,y)} \tag{3.10}$$

$$\mathrm{I}_{c_j}^{ga}(x,y,k) = \underset{x,y}{\operatorname{argmax}}[\bar{\mathrm{I}}_{c_j}^{bip}] + A(x,y))\frac{\mathrm{I}_{c_j}^{bip}(x,y,k)}{\mathrm{I}_{c_j}^{bip}(x,y,k) + A(x,y)}. \tag{3.11}$$

In the above equations, $H(x,y)$ and $A(x,y)$ are the adaptation factors of the horizontal cells and the amacrine cells respectively. These factors are both computed based on a weighted average (two-dimensional Gaussian) of surrounding pixel values plus an image-dependent global factor. For a more detailed description of this tone mapping operator, we refer to [38].

3.8 Post-Processing

The tone mapping operator described in Section 3.7.1 already performs a global tone correction. As in [38], we apply a gamma curve coupled with a luminance histogram stretching to the output image in Equation 3.11 to obtain the final, tone mapped HDR image:

$$\mathrm{I}_{c_j}^{\mathrm{TM}} = \mathrm{stretch\_hist}(\mathrm{I}_{c_j}^{\mathrm{ga}}, T_{\mathrm{low}}, T_{\mathrm{high}}, \frac{1}{\gamma}), \tag{3.12}$$

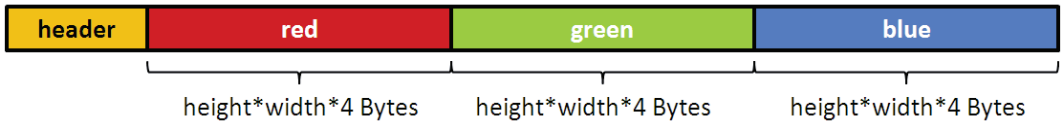
where stretch_hist(I, T_{low}, T_{high}) is a function which clips every value of the luminance channel of I that is less than T_{low} to T_{low}, and every value greater than T_{high} to T_{high}, and then applies a gamma curve whose shape is defined by γ together with a luminance histogram stretching. The value of γ is image dependent and defines the shape of the global tone mapping curve. For our ground truth image, we set $\gamma = 0.7$, $T_{\mathrm{low}} = 0.001$ and $T_{\mathrm{high}} = 0.999$.
The resulting tone mapped images show no halo artifacts and preserve the details in both dark and bright regions of the image. Figure 3.22 shows the right view of the tone mapped HDR image obtained by merging the 15 exposures of the ground truth sequence of the right camera.

3.9 Storage of HDR Radiance Map

In order to store the computed HDR radiance maps without losing any precision, we wrote a simple file format which consists of a header storing the most relevant details, together with the actual image data in floating point precision (4 Bytes). The *uEye* cameras used in this project capture color images at a resolution of 1,024x768 pixels (XGA). One image therefore needs $1,024 \cdot 768 \cdot 3 \cdot 4 = 9,437,184$ Bytes, excluding the header, which in the

Figure 3.22: Tone mapped HDR image resulting from merging the 15 exposures of the ground truth image sequence of the right camera.

current implementation is $6 \cdot 4 = 20$ Bytes. The total space required to store one frame is therefore $9,437,210$ Bytes.



(a) How the HDR raw content is stored.

(b) Content of the header.

Figure 3.23: Details on how the HDR image information is stored.

3.10 Conclusion

The whole workflow to get from a stereo exposure series to stereoscopic HDR content has been explained in this chapter. As expected, the camera response function has shown a highly linear behavior. While the exposure series alignment does not differ too much between the two approaches from a conceptual point of view, the output of this step has shown to be significantly worse for the spatial approach. The most problematic regions are regions that are at the same time clipped and occluded in one view. While clipped regions can be reliably detected by re-exposing one view to match the intensity values of the other, the fact that there is no information available due to occlusion leads to an unsolved problem. In the temporal approach, one can copy the information from the unclipped frame, assuming that the clipped region did not move.
For current stereoscopic displays, the images need to be tone mapped. A tone mapping operator that simulates parts of the non-linear processing of the HVS has been used, which gives visually pleasing results. It is important though that the raw radiance maps are also stored, as tone mapping is an irreversible operation. This way, the data is ready for future stereoscopic HDR displays.

4

Evaluation

The output produced in the processing step of the stereoscopic HDR pipeline is evaluated in this chapter. In the beginning, three full-reference metrics are presented and compared. The best one is then chosen to evaluate the quality of the HDR radiance map of a still scene for the left and right view independently. This is done for both the temporal and the spatial stereoscopic HDR approach. The focus is then put on the perceived quality of the HDR stereo pair. This is evaluated by looking at the tone mapped images on a stereoscopic display. Stereoscopic HDR video is also evaluated using visual evaluation on a stereoscopic display.

Figure 4.1: Parts of the evaluation block.

4.1 Image Quality Metrics Comparison

Image quality metrics aim at evaluating the perceived quality of an image. *Full-reference metrics* are metrics that take the (distorted) test image and compare it with an undistorted reference (ground truth) image. In our case, this is the HDR radiance map created by combining all 15 images of the ground truth exposure series. We chose to compare three image quality metrics, namely the *peak signal to noise ratio (PSNR)*, the *structural similarity (SSIM)* by Wang *et al.* [43] and the HDR VDP 2.0 by Mantiuk *et al.* [21]. The reason for this choice is that the PSNR is the de-facto standard to measure the quality of an image. It is well known however that this metric does not correlate too well with perceived quality due to the non-linearity of the HVS, which is not taken into account

in the computation of the PSNR. This is why we considered two other quality metrics which take those non-linearities into account. In a recent comparison of image quality metrics by Ponomarenko *et al.* [30], the SSIM has been shown to give very good results. The last metric is the HDR VDP 2.0, which is the most complex metric of the three, and is expected to give the best results. In the following, the three metrics are explained and then compared in order to find the metric that correlates best with what a visual inspection of the outputs would give.

4.1.1 Peak Signal to Noise Ratio (PSNR)

The PSNR measures the ratio between the maximum possible power that can be in a signal and the amount of noise that is present in the test image as compared to the ground truth image. The amount of noise is computed by the *mean squared error (MSE)*:

$$\mathrm{MSE} = \frac{1}{h \cdot w} \sum_{x=1}^{h} \sum_{y=1}^{w} \left[\mathrm{I}_{c_j}^{\mathrm{TM_{test}}}(x,y) - \mathrm{I}_{c_j}^{\mathrm{TM_{groundtruth}}}(x,y) \right]^2, \tag{4.1}$$

where h and w are the height and the width of the input images respectively. The PSNR, measured in dB, is then simply computed as:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{L_{\max}^2}{\mathrm{MSE}} \right), \tag{4.2}$$

where $L_{\max}$ is the maximum representable intensity value, i.e. the peak signal value. In our case, $L_{\max} = 255$.

4.1.2 Structural Similarity (SSIM)

This quality metric by Wang *et al.* [43] was designed to improve upon the PSNR. This measure separates the task of measuring similarity into three comparisons, namely *luminance*, *contrast* and *structure*. A detailed description on how these three comparisons are performed can be found in [43]. To adhere to the notations in [43], let $\mathbf{x} = \mathrm{I}_{c_j}^{\mathrm{TM_{test}}}$ and $\mathbf{y} = \mathrm{I}_{c_j}^{\mathrm{TM_{groundtruth}}}$. The combination of the three comparisons takes, after simplification, the following form:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{4.3}$$

where μ_x denotes the mean of $\mathbf{x}$, σ_x^2 denotes the variance of x, σ_{xy}^2 denotes the covariance of $\mathbf{x}$ and $\mathbf{y}$, and C_1 and C_2 are constants in order to avoid unstable results that arise if either $\mu_x^2 + \mu_y^2$ or $\sigma_x^2 + \sigma_y^2$ are close to 0.

4.1.3 High Dynamic Range Visible Difference Predictor (HDR VDP) 2.0

Mantiuk *et al.* [21] proposed an objective quality metric to predict visible differences in HDR images. The output of this method is a probability map of visible differences, as well as a score between 0 and 100 indicating how severe the distortion is, denoted as Q_{MOS}.

It is shown that the Q_{MOS} is a good alternative to *multi-scale structural similarity (MS-SSIM)* for applications that need finer control over viewing parameters such as display brightness and viewing distance. On top of that, it can also be used to measure the quality for scenes that go beyond the luminance range of typical LCD or CRT displays. The interested reader is referred to [21] to find all the details of the computation of the Q_{MOS}. Here, we just give the final formula:

$$Q_{\text{MOS}} = \frac{100}{1 + \exp(q_1(Q + q_2))}. \tag{4.4}$$

In the above equation, Q is the pooled information of the visible difference probability map computed by the HDR VDP 2.0.

4.1.4 Selection of best Quality Metric

In order to find out which quality metric gives the best results, we chose all possible combinations of two out of the 15 exposures from the ground truth exposure series and merged them to HDR radiance maps, as shown in Figure 4.2. Since both PSNR and SSIM are limited to LDR images, we applied both the tone mapping explained in Section 3.7.1 and post-processing (Section 3.8) to the radiance maps.

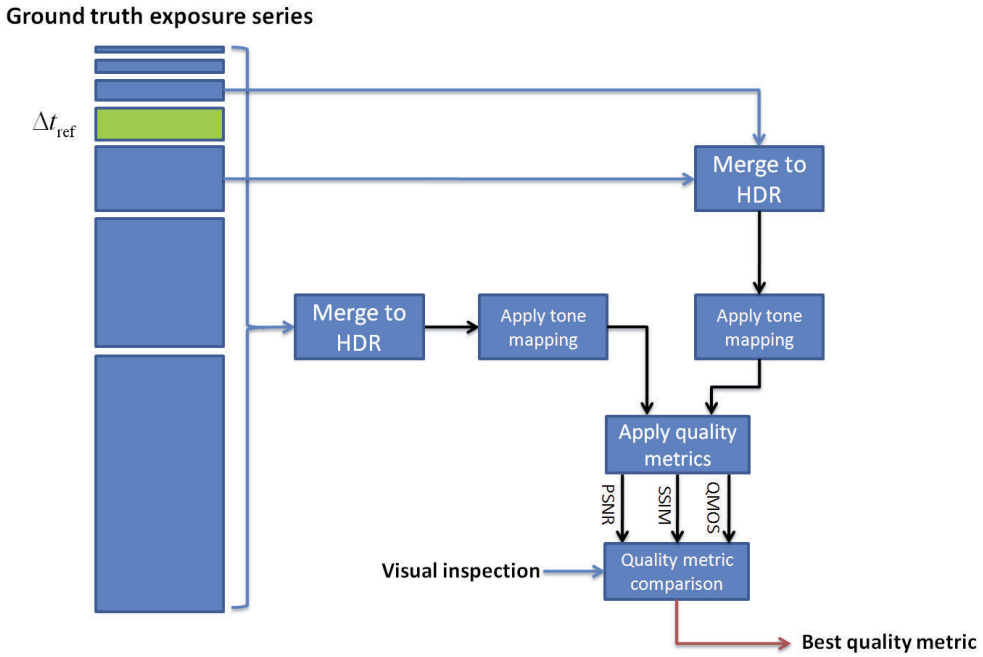
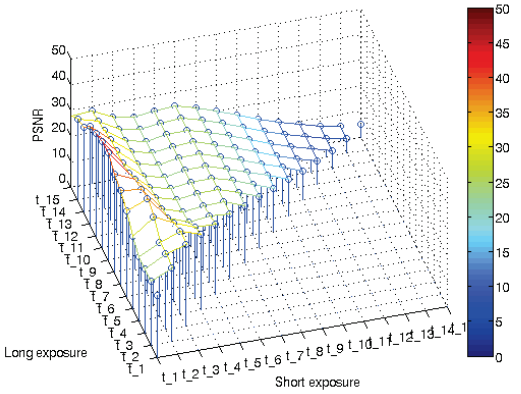


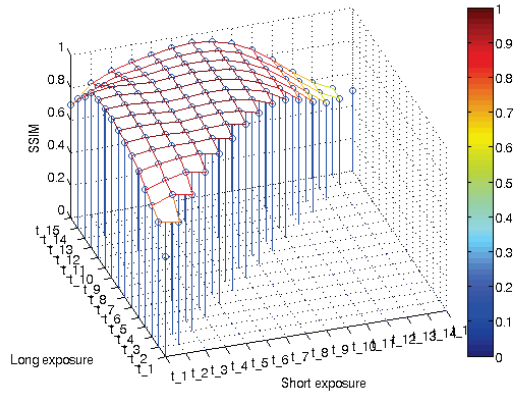
Figure 4.2: How the quality metric used for the rest of the experiments was selected.

While the SSIM and the HDR VDP 2.0 have an upper limit which is 1.0 and 100.0 respectively, the PSNR has no upper bound. In order to put the three metrics on comparable scales, we set the maximum PSNR value to 50 dB, which is considered a very good quality. Figure 4.3 shows the plots for the three quality metrics.
Since the results for the left and the right camera were almost identical for all three metrics, we only show the plots for the exposures of the left camera, along with a cropped part of the tone mapped HDR image that was predicted to have the highest perceived



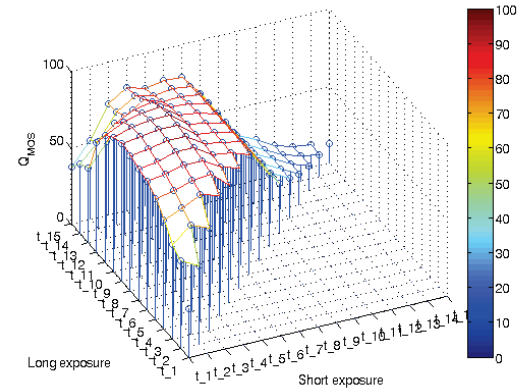
(a) PSNR, peak value of 41.77 dB for the pair $(\Delta t_1, \Delta t_9)$.

(b) Cropped, tone mapped result for the pair $(\Delta t_1, \Delta t_9)$.



(c) SSIM, peak value of 0.9795 for the pair $(\Delta t_2, \Delta t_{10})$.

(d) Cropped , tone mapped result for the pair $(\Delta t_2, \Delta t_{10})$.



(e) HDR VDP 2.0, peak value of $Q_{\mathrm{MOS}} = 94.03$ for the pair $(\Delta t_4, \Delta t_{10})$.

(f) Cropped , tone mapped result for the pair $(\Delta t_4, \Delta t_{10})$.

Figure 4.3: Comparison of the three metrics.

quality for each of the three metrics. We can see that the PSNR achieves the highest values for combinations of darker images, and that the resulting tone mapped image contains a lot of noise. The shapes of the curves obtained for SSIM and HDR VDP 2.0 look similar, although the one for SSIM is a bit less discriminative. When looking at the resulting tone mapped images obtained for the peak exposures, they both look reasonable.
Because of the fact that HDR VDP 2.0 can incorporate a display model as well as its ability to predict visible differences in HDR radiance maps, we chose the Q_{MOS} as quality metric to evaluate the perceived quality of the created HDR radiance maps.

4.2 HDR Image Quality

The first part of the evaluation is based on still scenes, and is aimed at evaluating the quality of the HDR radiance maps for the left and the right view individually. All the experiments are carried out using the ground truth scene shown in Figure 3.2. This scene was selected because it presents a typical case where HDR is needed.
As mentioned before, we use the Mantiuk *et al.*'s HDR VDP 2.0 to compute the perceived quality of the generated HDR radiance maps. This metric is used to find out which two captures best represent the whole dynamic range of the scene, both for the temporal and the spatial stereoscopic HDR. Driven by the fact that one can easily add more exposures to the temporal stereoscopic HDR approach, we also investigate the best trade-off between temporal sampling frequency and increase in dynamic range. The HDR VDP 2.0 also allows to model a display. Our display model is explained in the following.

4.2.1 Display Model

The HDR VDP 2.0 needs the color encoding to be specified explicitly. For our purposes, the XYZ color space was best suited. Since the HDR radiance maps are computed in RGB color space, the ground truth and test images had to be converted, as shown in Figure 4.4.

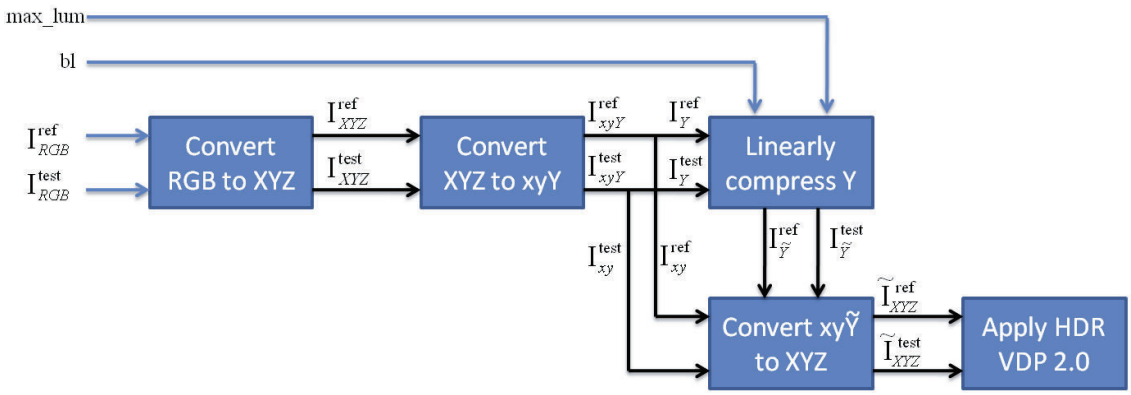


Figure 4.4: How the radiance map is converted to XYZ with desired black level bl and maximum luminance max_lum.

First, the HDR radiance map is converted to XYZ color space. We assume sRGB [36] for the RGB working space, which defines the RGB to XYZ transformation matrix M as:

$$\mathrm{M} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \tag{4.5}$$

The next step is to convert XYZ to xyY. The *luminance channel* Y is then linearly mapped to match the desired black level (bl) and maximum luminance value (max_lum) of the display. The conversion from the HDR radiance map to luminance values has been done using the following equation:

$$Y_{\text{mapped}} = \frac{Y \cdot (max\_lum - bl)}{Y_{\max}} + bl, \tag{4.6}$$

where $Y_{\max}$ corresponds to the highest value of Y in the ground truth radiance map. The HDR display modelled for the evaluation has a black level of 0.03 cd/m^2 and a maximum luminance value of 2500 cd/m^2, which gives a dynamic range of 83333:1. The size of the display is 22" with a resolution of (1680x1050), at a viewing distance of 0.6 m.

4.2.2 Temporal HDR - Two out of 15

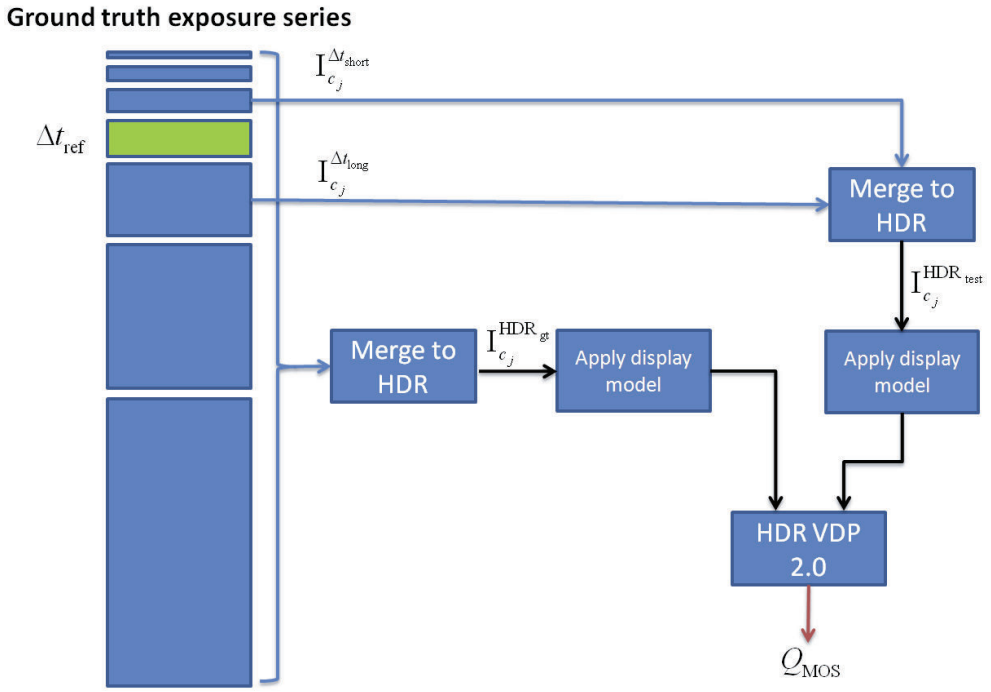
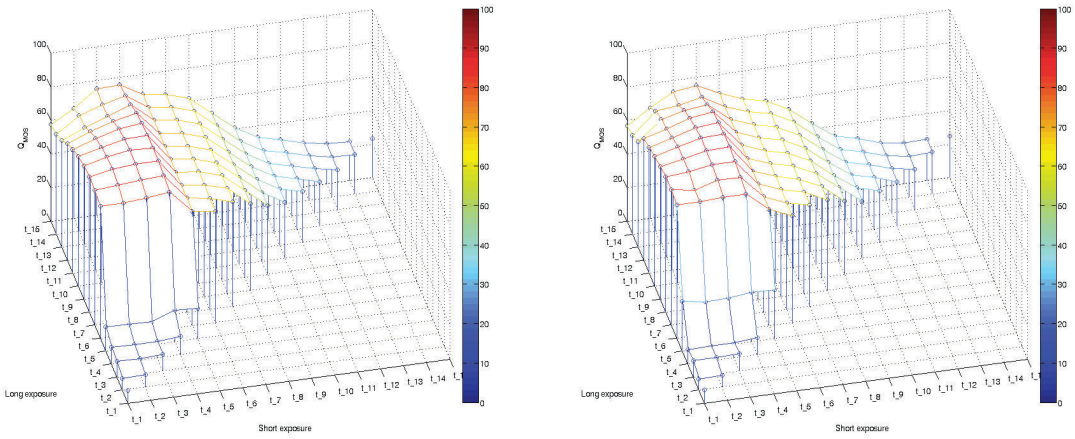


Figure 4.5: How the ground truth and test HDR images are being created for the temporal approach. The green frame shows the reference frame. Note that for visualization purposes, the ground truth image only consists of $N_{\text{exp}} = 7$ different exposures.

In order to answer the question which two out of the 15 captures of the ground truth exposure series would give the best result in terms of visual quality, we tried out all the

combinations where the short exposure time was shorter than the long exposure time. Figure 4.6 shows the plot of the Q_{MOS} computed.



(a) Left view (b) Right view

Figure 4.6: Q_{MOS} computed by the HDR VDP for all the combinations of short and long exposure times. The numbers on the axis refer to the exposure times shown in Figure 3.2. The peak Q_{MOS} of 94.71 and 94.46 for the right and left camera is achieved for the pair $(\Delta t_4, \Delta t_{11})$, corresponding to an exposure ratio of $e = 8\sqrt{2}$.

We can see that the maximum Q_{MOS} is achieved for the pair $(\Delta t_4, \Delta t_{11})$ for both cameras. This makes sense, since Δt_4 is the longest exposure time at which the clouds outside are still captured with great detail, but due to the longer exposure time than for $\Delta t_1 - \Delta t_3$, there is less noise in the dark parts. The choice of the longer exposure is a harder one, as it is more difficult to tell at which point enough details in the dark areas are captured. It is interesting to compare the results obtained in this experiment with the ones obtained in Section 4.1.4, where the Q_{MOS} was computed on the tone mapped version. The shape of the two curves are quite similar. One can note that the Q_{MOS} on the HDR radiance maps falls a bit more for exposure combinations consisting of two quite short exposure times. This may be due to the fact that our display model linearly maps the radiance values to luminance, as can be seen in Equation 4.6.

4.2.3 Spatial HDR - Two out of 15

It is clear that when applied on an exposure series of a static scene, the results for the temporal approach will always be better than the ones for the spatial approach. In fact, a static scene implies that all images in the temporal approach are perfectly aligned. What we wanted to find out was by how much worse the results are for the spatial approach. For this, we chose a similar setup as for the previous experiment (see Figure 4.7).
The difference to the evaluation of the temporal approach is that we select the short exposure from the ground truth of the right camera, and the long exposure from the ground truth of the left image. For evaluating the quality of the right HDR view, we apply the image alignment procedure as explained in Section 3.4.4 on the left image, and

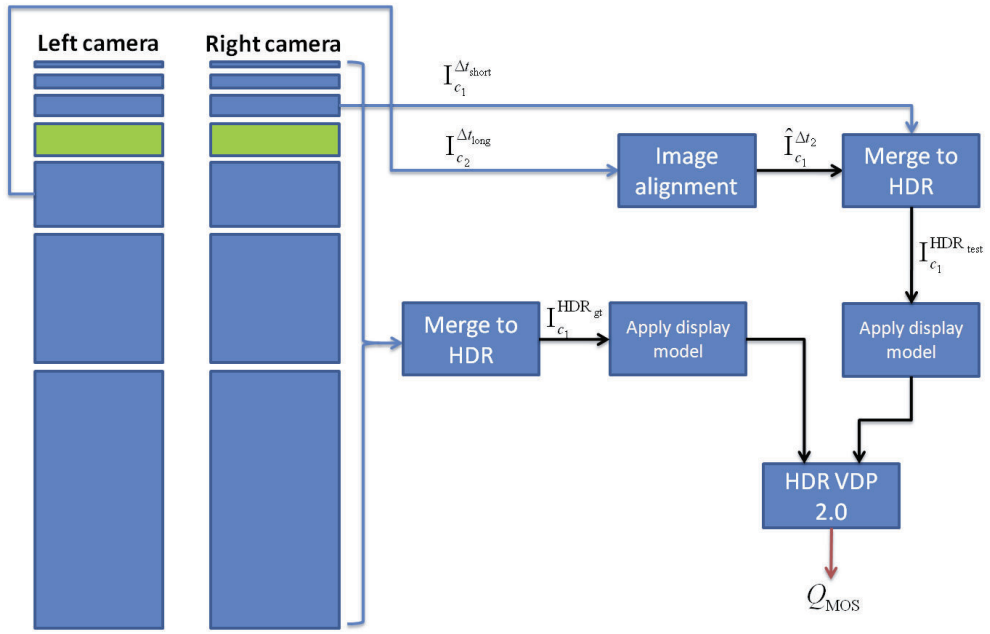
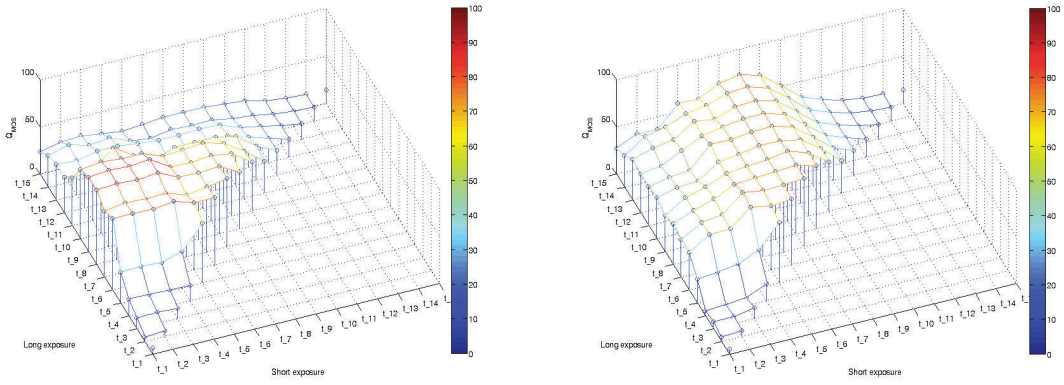


Figure 4.7: Pipeline to get the ground truth and test HDR images for the spatial method. Note that for visualization purposes, the ground truth image only consists of $N_{\text{exp}} = 7$ different exposures.

then compare it to the ground truth HDR radiance map. Figure 4.8 shows the plot of the computed Q_{MOS} for all possible pairs of images where $\Delta t_{c_1} \leq \Delta t_{c_2}$.



(a) Left view (b) Right view

Figure 4.8: Q_{MOS} computed by the HDR VDP 2.0 for all the combinations of short and long exposure times. The numbers on the axis refer to the exposure times shown in Figure 3.2.

Comparing these results to the one obtained for the temporal approach (see Figure 4.6), two things are immediately apparent. First, as expected, the results are significantly worse than for the temporal approach. What is maybe less obvious is that the results for the left and the right camera are less consistent than for the temporal approach, where the results for the left and the right view are almost identical. Therefore, it is much more

difficult to find the two exposure times in the spatial approach.

4.2.4 Temporal HDR - *n* out of 8

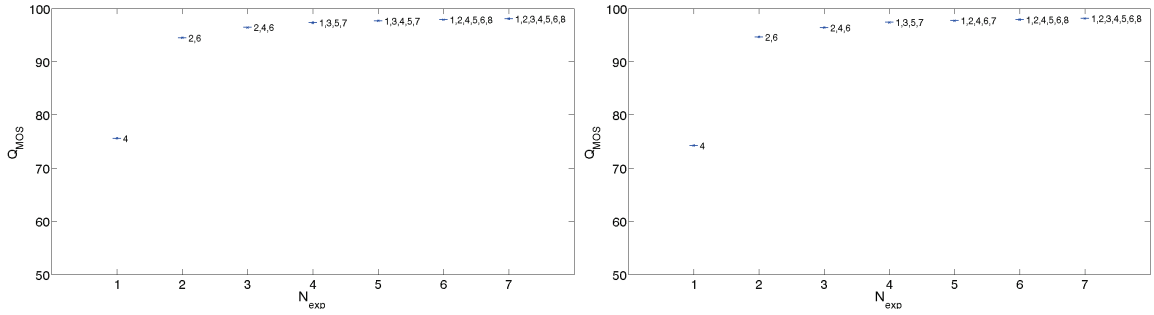
As mentioned earlier, in the temporal approach one can easily trade off temporal sampling frequency with extension in dynamic range by using $N_{\text{exp}} > 2$ exposure times. The interest of this test is to find out at which point adding more exposure times does not result in an important gain in perceived quality of the output image. What we are interested in are all possible combinations of $N_{\text{exp}} = 1$ up to 14 exposures and their comparison to the HDR ground truth image obtained by using the 15 exposures. Unfortunately, the number of possible pairs is too large. In fact, the number of possible pairs is equal to:

$$\sum_{k=1}^{n} \binom{n}{k} = \sum_{k=1}^{15} \binom{15}{k} = 2^{15} - 1 = 32767. \tag{4.7}$$

In order to make the problem of finding the best trade-off between number of captures used in the exposure series - which we want to keep as low as possible - and the perceived gain in image quality more practical, we decided to select every second image of the exposure series, i.e. to only selecting the exposures corresponding to the exposure times $\Delta t_1, \Delta t_3, ..., \Delta t_{15}$. This way, the total number of possible pairs is reduced to:

$$\sum_{k=1}^{n} \binom{n}{k} = \sum_{k=1}^{8} \binom{8}{k} = 2^{8} - 1 = 255. \tag{4.8}$$

In this experiment, all possible permutations going from one to eight captures from the ground truth scene were tested. We refer to this new set of images as *reduced ground truth*. Figure 4.9 shows the highest obtained Q_{MOS} for each number of exposures.



(a) Left camera (b) Right camera

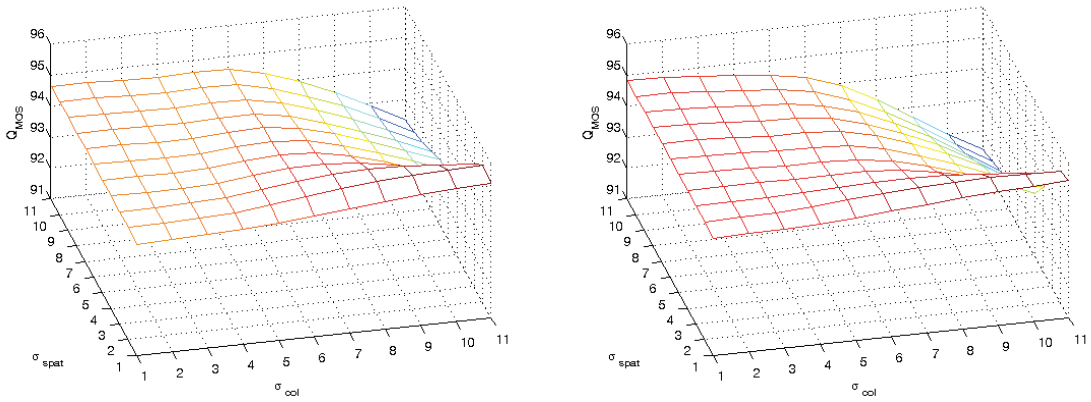
Figure 4.9: Highest Q_{MOS} for each value of N_{exp}, computed by the HDR VDP 2.0. The numbers next to the achieved Q_{MOS} correspond to the exposures of the reduced ground truth set. We can see that starting from three captures, the value of the Q_{MOS} almost stagnates, showing that three captures are a good trade-off between number of captures and increase in dynamic range.

We can see that image 4 (corresponding to exposure time Δt_7) would give the highest Q_{MOS} if only one capture could be selected. This confirms that the reference exposure is chosen correctly. For two captures, the best result was obtained by combining images 2 and 6 (corresponding to exposure times Δt_3 and Δt_{11}), which confirms the result obtained

in Section 4.2.2. For three captures, exposures 2, 4 and 6 give the highest Q_{MOS}. It has to be noted that adding a third exposure can substantially increase the time it takes to capture one HDR frame. For the scene under scope, the total exposure time using two captures is $\Delta t_3 + \Delta t_{11} = 33.98$ ms. For three exposures, the total time to capture one frame with three exposures is $\Delta t_3 + \Delta t_7 + \Delta t_{11} = 41.95$ ms, which is around 1.25 times the time it takes for $N_{\text{exp}} = 2$. The situation gets even worse for $N_{\text{exp}} > 3$, and the resulting Q_{MOS} does not significantly improve. Only the variance of the obtained Q_{MOS} gets smaller as N_{exp} increases.

4.2.4.1 Noise Reduction by Filtering the HDR Radiance Map

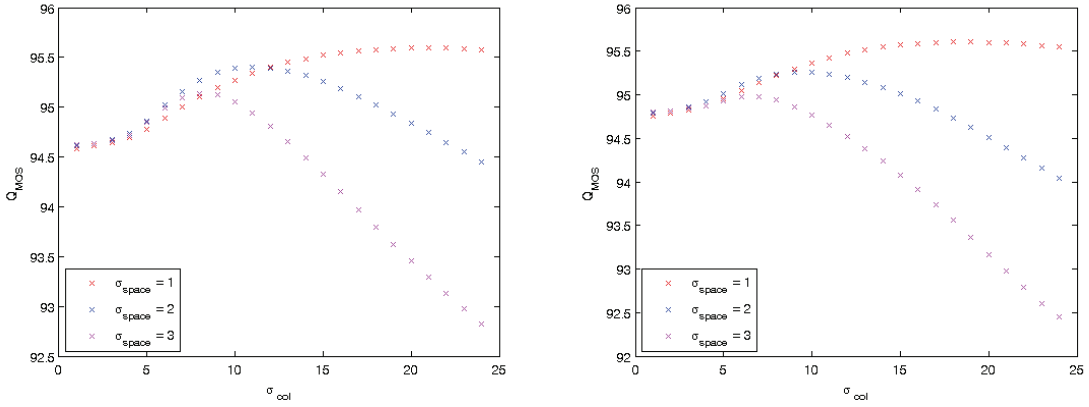
We wanted to find out whether the Q_{MOS} could be improved by applying a noise reduction filter on the HDR radiance map having the highest Q_{MOS} for $N_{\text{exp}} = 2$, namely the HDR image obtained by merging the images corresponding to exposure times Δt_3 and Δt_{11}. Bilateral filters have been shown to be effective at reducing noise while preserving the edges in an image [29]. The bilateral filter implementation of OpenCV 2.1 has been used for this experiment, which takes, besides other arguments, the sigma for the color and the sigma for the spatial component, denoted as σ_{col} and σ_{space}, respectively, in the following. All possible combinations of $\sigma_{\text{col}} \in \{1, 2, ..., 11\}$ and $\sigma_{\text{space}} \in \{1, 2, ..., 11\}$ have been applied to the HDR radiance map of the image pair (2,6), which was the one which yielded the highest Q_{MOS} for $N_{\text{exp}} = 2$. The results are shown graphically in Figure 4.10.



(a) Left camera (b) Right camera

Figure 4.10: Results of applying a bilateral filter for different filter sizes to the HDR radiance map of captures 2 and 6 of the reduced ground truth set.

We can see that in general, the Q_{MOS} is slightly improved, and that the biggest improvements are made for small spatial filter sizes. Inspired by these results, more filter sizes have been tested for $\sigma_{\text{col}} \in \{1, 2, ..., 24\}$ and $\sigma_{\text{space}} \in \{1, 2, 3\}$. The results are shown in Figure 4.11.
Getting back to the initial values of the Q_{MOS} for the pair (2,6), where we had $Q_{\text{MOS}} = 94.64$ and $Q_{\text{MOS}} = 94.49$ for the right and left view respectively, the results are improved by around 1% ($Q_{\text{MOS}} = 95.61$ and $Q_{\text{MOS}} = 95.60$ for the right and left view respectively),



(a) Left camera (b) Right camera

Figure 4.11: Q_{MOS} obtained for different combinations of filter sizes. We can see that the best results are obtained for a spatial filter size $\sigma_{\text{space}} = 1$. The highest Q_{MOS} is obtained for $\sigma_{\text{col}} = 19$ and $\sigma_{\text{space}} = 1$ with a value of $Q_{\text{MOS}} = 95.61$ for the right camera, and for $\sigma_{\text{col}} = 21$ and $\sigma_{\text{space}} = 1$ with a value of $Q_{\text{MOS}} = 95.60$ for the left camera.

and get very close to the maximum Q_{MOS} obtained using three exposures, which were at $Q_{\text{MOS}} = 96.38$ and $Q_{\text{MOS}} = 96.42$.

4.3 Quality of the Stereo Pair

While the HDR VDP 2.0 provides a framework to automatically evaluate the perceived quality of HDR, it does not tell anything about the perceived quality of the stereo image pair. This aspect will be analyzed in this section based on visual evaluation on a stereoscopic display. More precisely, a 3D ready *Philips 9000 LED series* was used. In combination with the *Philips 3D upgrade kit*, this TV can show stereoscopic content using active shutter glasses. We start by looking at the results for the temporal stereoscopic HDR and then evaluate the quality of the spatial stereoscopic HDR.

4.3.1 Temporal Stereoscopic HDR

In the previous section, we have found the best choices for the exposure times of the left and the right camera. Furthermore, we have seen that for the temporal approach, the quality of the HDR radiance map obtained using two out of the 15 exposures is only slightly worse than when using three out of 15, but that the total amount of time required to capture three frames is much shorter. We now look at the stereo pairs of the best choices as selected by the HDR VDP 2.0.
Figure 4.12 shows the tone mapped stereo pairs that gave the highest Q_{MOS} for $N_{\text{exp}} = 2$, $N_{\text{exp}} = 3$ and for the *reduced ground truth set*. We further refer to them as *stereo pair* 1, 2 and 3, respectively. First of all, it can be noted that the left and the right view are consistent and that there no disturbance between the left and the right view. Comparing the perceived quality of the three stereo pairs, it can be noted that there is no visible difference between *stereo pair* 1 and 2. *Stereo pair* 3 contains the shortest exposure of

(a) Left view for (Δt_3, Δt_{11}) (b) Right view for (Δt_3, Δt_{11})

(c) Left view for (Δt_3, Δt_7, Δt_{11}) (d) Right view for (Δt_3, Δt_7, Δt_{11})

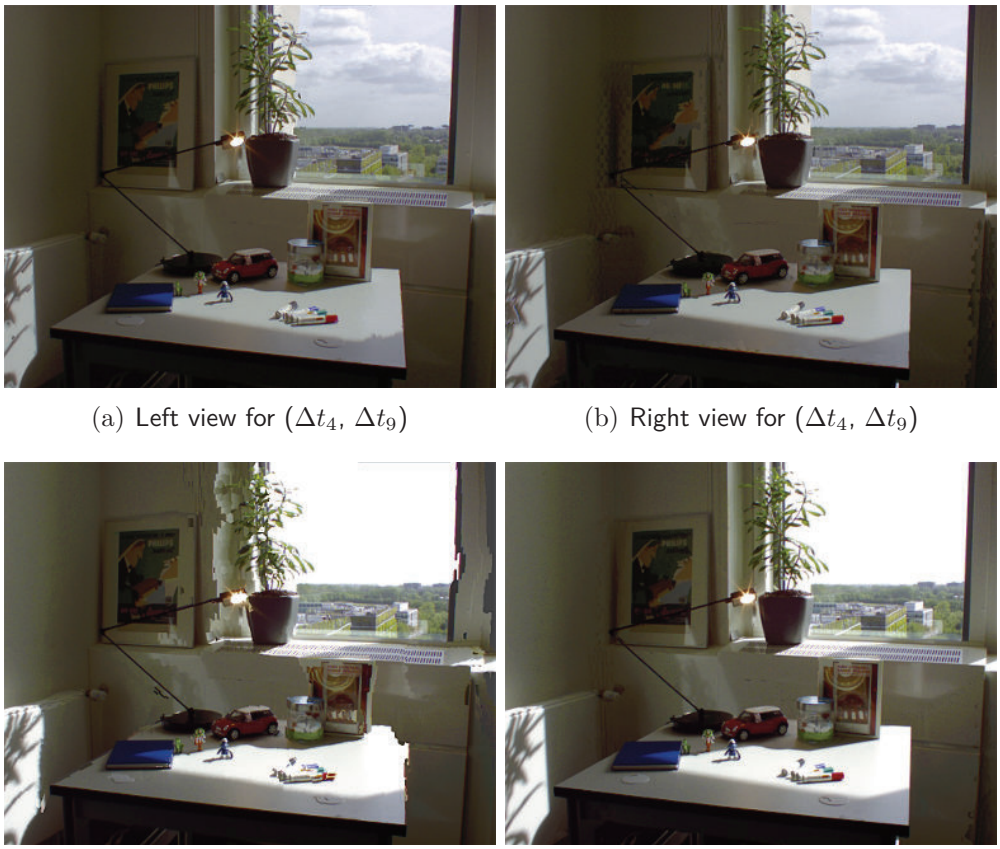
(e) Left view for *reduced ground truth set* (f) Right view for *reduced ground truth set*

Figure 4.12: Tone mapped stereo HDR pairs as selected to be the best ones by the HDR VDP 2.0 for the temporal approach.

the exposure series, which is why there are more details in the lamp. In fact, one can almost see the filament of the lamp. When looking at the real scene, this filament can only hardly be perceived, which is why it may look unnatural and even be unwanted.

4.3.2 Spatial Stereoscopic HDR

As we have already seen in Section 4.2.3, the highest Q_{MOS} was not obtained for the same pair of exposures. We therefore selected the pair which gave the highest Q_{MOS} for the left view, and the one which gave the highest Q_{MOS} for the right pair. They are shown in Figure 4.13.



(a) Left view for (Δt_4, Δt_9) (b) Right view for (Δt_4, Δt_9)

(c) Left view for (Δt_7, Δt_{14}) (d) Right view for (Δt_7, Δt_{14})

Figure 4.13: Tone mapped stereo HDR pairs as selected to be the best ones by the HDR VDP 2.0 for the spatial approach.

One can immediately see that the result will be inconsistent on the display. While the result for pair (Δt_4, Δt_9) is quite pleasing to view, the one for the stereo pair (Δt_7, Δt_{14}) is impossible to be fused. One thing that is interesting to note is that for the sky part of the pair (Δt_4, Δt_9) , the iterative disparity propagation algorithm has worked quite well. The sky appears a bit closer to the screen than in the ground truth scene, but this does not further influence the viewing experience. The pair (Δt_7, Δt_{14}) then gives a good example where the IDP fails as there are not enough reliable disparities adjacent to the clipped regions.

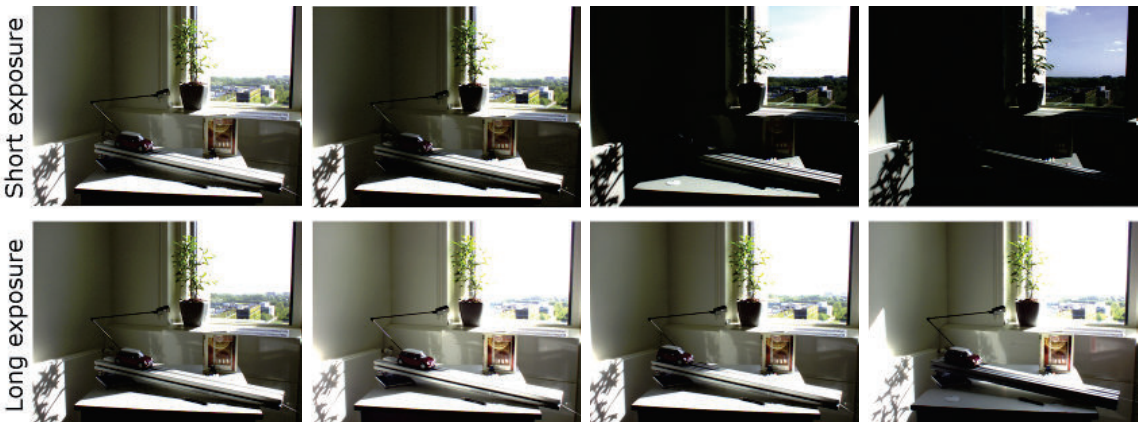
4.4 Stereoscopic HDR Video

The temporal stability of the two modes is evaluated based on the car scene presented in Section 3.5. It consists of a toy car driving down a slope created using two metal bars,

recorded with different exposure ratios.

4.4.1 Temporal Stereoscopic HDR

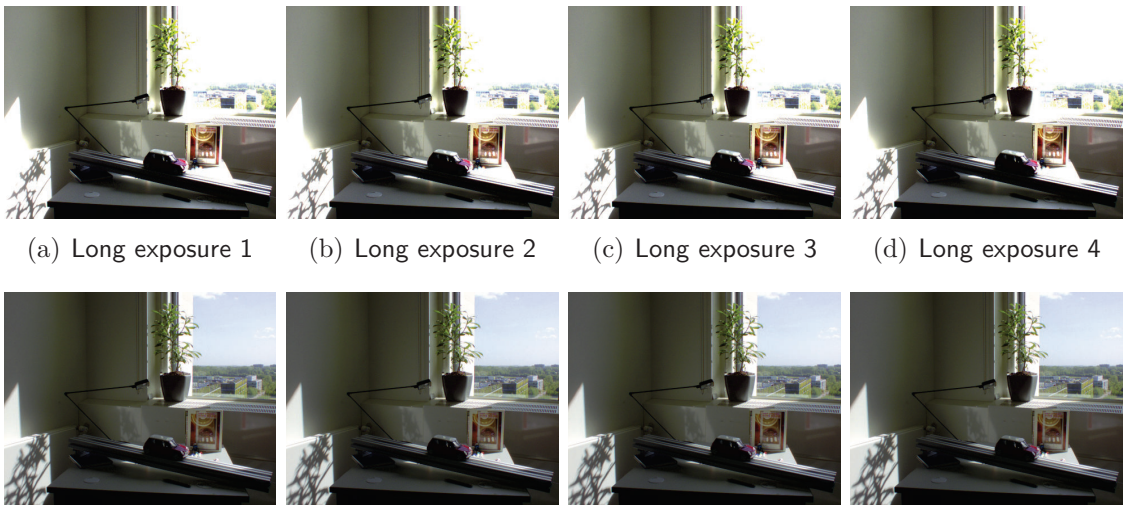
This section evaluates the temporal approach based on the recorded car scene. This scene has been recorded several times with different exposure ratios. We used exposure ratios $e = 1$, $e = 2$, $e = 4$, and $e = 8$. For the exposure ratio of $e = 1$, we set the exposure time to the reference time Δt_{ref}, and for the other cases they are set such that $\Delta t_{short} < \Delta t_{\text{ref}} < \Delta t_{long}$, and $\Delta t_{long} = e \cdot \Delta t_{short}$. Figure 4.14 shows the first two frames recorded for the different exposure ratios.



(a) $e = 1$. (b) $e = 2$. (c) $e = 4$. (d) $e = 8$.

Figure 4.14: First two frames of the car scene recorded by the right camera, for different exposure ratios e.

In order to avoid ghost artifacts in regions where there is motion, the motion compensation algorithm presented in Section 3.5 is applied to align the exposures. The higher the exposure ratio, the higher the amount of clipped regions, and the harder the motion estimation. On the other hand, a larger exposure ratio results in a higher increase in dynamic range.
Looking at the resulting stereoscopic HDR video on a stereoscopic display, several observations can be made. First, as mentioned before, the tone mapped left and right view are consistent for all tested exposure ratios. Second, the very simple motion estimation algorithm is able to cope with all exposure ratios and there are only few artifacts, even for $e = 8$. Since the gain in dynamic range is the highest for $e = 8$, one can conclude that this exposure ratio gives the most satisfying results. It is interesting to compare the video of the long exposures of the recording for $e = 8$ with the tone mapped HDR video. Figure 4.15 shows sample frames of the two generated videos. While the long exposure video captures the details inside, the tone mapped HDR video captures both details inside and outside. The result is a much more natural looking scene, which gives a better viewing experience.



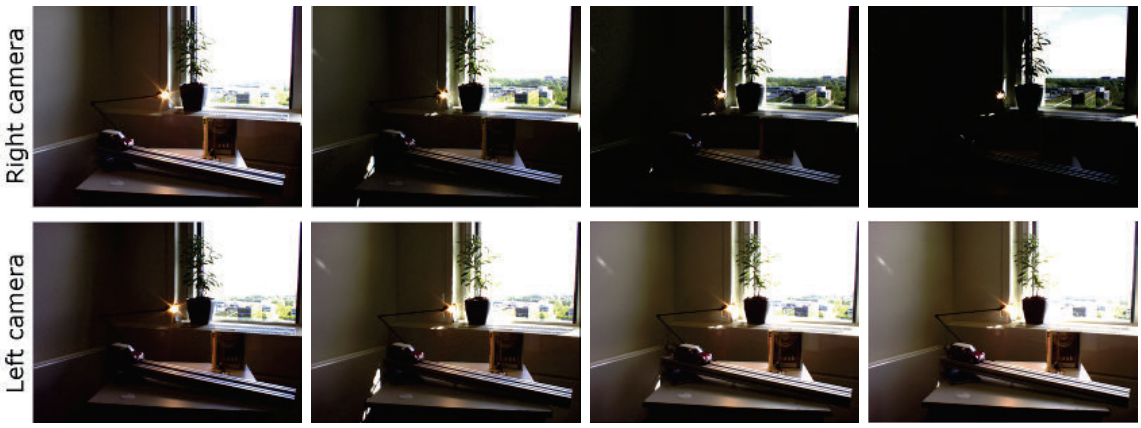
(a) Long exposure 1 (b) Long exposure 2 (c) Long exposure 3 (d) Long exposure 4

(e) Tone mapped 1 (f) Tone mapped 2 (g) Tone mapped 3 (h) Tone mapped 4

Figure 4.15: 4 sample frames of the right camera. *Top:* Long exposures, capturing the details inside, but outside with clipped sky. *Bottom:* Tone mapped frames, capturing details both inside and outside.

4.4.2 Spatial Stereoscopic HDR

As for the temporal mode, the car scene has been recorded at different exposure ratios. The first frames recorded are shown in Figure 4.16.

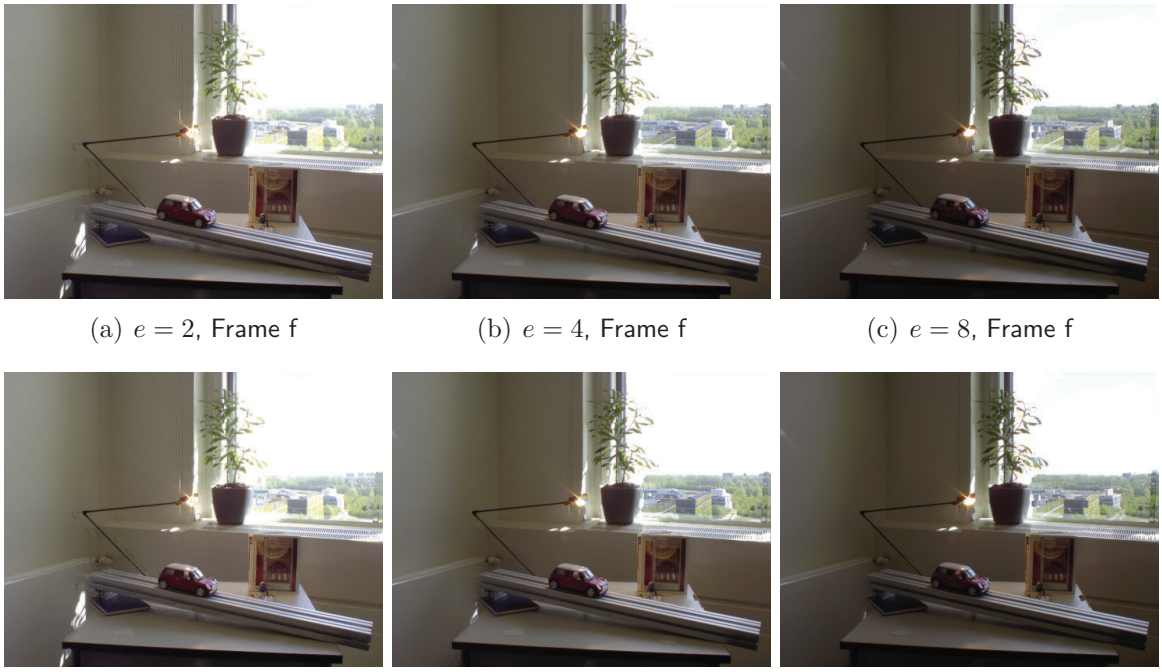


(a) $e = 1$. (b) $e = 2$. (c) $e = 4$. (d) $e = 8$.

Figure 4.16: First frame of the car scene recorded by the right and left cameras, for different exposure ratios e.

Note that the long exposure time was set high enough that details are visible inside. This resulted in the fact that even in the short exposure, the sky is almost not visible, even for an exposure ratio $e = 8$.
As mentioned before, the clipped region handling is more difficult for the spatial approach than for the temporal approach. For the temporal approach, the assumption that the clipped regions did not move allowed copying of the data from the next frame, where the region is not clipped. In the case of spatial stereoscopic HDR, this is not possible

as the information needs to be fetched from the other camera. The iterative disparity estimation algorithm proposed in Section 3.4.3.1 computes the disparity of clipped regions based on neighbors with valid disparities. The larger the exposure ratio, the more clipped regions there are. This in turn implies that the estimated disparities will be less and less accurate. On top of that, if the image segmentation is not identical, the neighbors of a clipped region will never be exactly the same, leading to more and more artifacts in the clipped regions. The tone mapped results of two consecutive frames for the tested exposure ratios are shown in Figure 4.17.



(a) $e = 2$, Frame f (b) $e = 4$, Frame f (c) $e = 8$, Frame f

(d) $e = 2$, Frame f+1 (e) $e = 4$, Frame f+1 (f) $e = 8$, Frame f+1

Figure 4.17: Two consecutive frames for different exposure ratios of the left camera.

The regions that are in both views not clipped look fine in the tone mapped frames for all examined exposure ratios. In particular, the moving car is perfectly aligned. In the clipped regions, the temporal stability deteriorates, leading to parts that seem to be jumping. Note that this becomes even more disturbing when looking at the stereoscopic pair, as these regions will be correct in the other view.

4.5 Conclusion

In this chapter, the HDR radiance maps produced in the processing step of the stereoscopic HDR pipeline have been evaluated. Since there is no metric that is able to evaluate the quality of stereoscopic HDR, the evaluation has been done in several parts.
In the beginning of the chapter, three full reference metrics have been compared. This evaluation has yet once more shown that the PSNR is not a very reliable metric to measure perceived image quality. The SSIM and the HDR VDP 2.0 (Q_{MOS}) yielded results that correlated much better with the visual evaluation of the tone mapped radiance maps. The computation of the SSIM is much faster than the one of the Q_{MOS}, which makes it

better suited for time-critical applications. In our case, time was not an issue. Because of its ability to predict the perceived quality of HDR images, the HDR VDP 2.0 was the metric of choice for the rest of the evaluation.
In the first part of the evaluation of the quality of the stereoscopic HDR radiance maps, all possible combinations of two out of the 15 ground truth exposures were chosen and merged to an HDR radiance map. Not surprisingly, the temporal approach gave better results than the spatial one. What was unexpected was that in the spatial approach, the results giving the highest perceived quality for the left view were obtained for quite different pairs of exposures than for the right view. This makes it much harder to find a good pair of exposures which gives a consistent left and right stereo pair.
Another interesting result is that the scene at hand which has quite a large dynamic range can be very well approximated by using only two carefully chosen exposures. This result was confirmed by visual evaluation of the tone mapped HDR images on a stereoscopic screen. This screen also allowed to evaluate the consistency between the left and the right view.
In the last part, the temporal stability has been evaluated. For the recorded scene that contained moving objects, the simple region-based motion estimator used in the case of temporal stereoscopic HDR successfully compensated the motion and lead to an almost artifact free video. Note however that for a scene with more complex motion, the simple motion estimator would likely go wrong. In the spatial approach, the clipped regions had quite a lot of artifacts due to wrongly estimated disparity vectors, which is a problem even a more advanced disparity estimator could not solve.

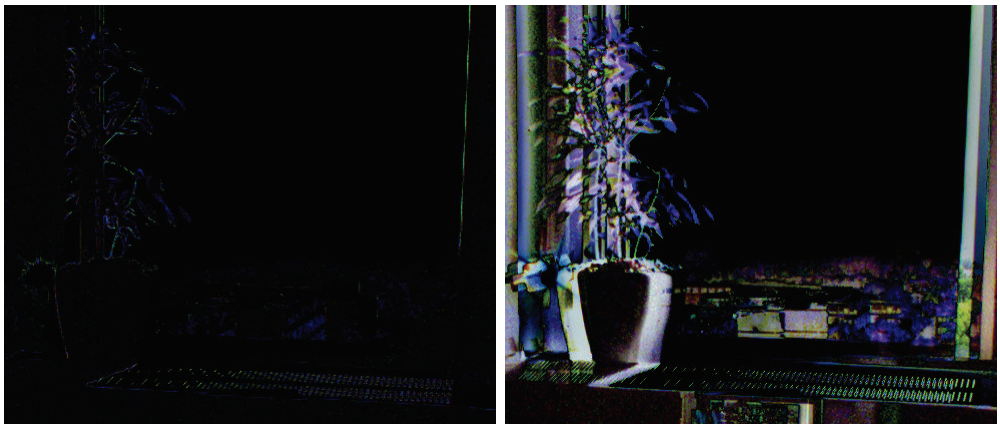
5

Conclusions and Future Work

In this thesis, a method for the recording, processing, and evaluation of stereoscopic HDR content has been proposed and implemented.
In the first part of the stereoscopic HDR pipeline, two different recording modes have been presented, referred to as temporal and spatial approaches due to the nature they operate. The advantage of the spatial approach is that it allows to capture at a high frame rate equal to the longer of the two fixed exposure times. Also, the two cameras capture at the same instant in time, which is not the case in the temporal approach, where both cameras are alternate between short and long exposure time, and the frame rate equals the sum of the exposure times. This is an advantage for the spatial mode since changing the exposure time is time-consuming and further lowers the frame rate of the temporal mode.
The processing part of the pipeline has shown several disadvantages of the spatial approach. In fact, the information exchange between the left and the right camera based on disparity estimation is already difficult for images taken at the same exposure time. Changing the exposure times introduces *half-clipping*, i.e. regions that are clipped in one view and not in the other. This becomes an issue since the left and the right view eventually need to be consistent. The temporal approach has the clear advantage here that the images used to create the HDR radiance map are all coming from the same camera, which removes the need for realignment.
A thorough evaluation of the obtained HDR radiance maps has shown that an HDR still scene can be well approximated using two well-chosen exposures. Applying a bilateral filter on the radiance map for noise reduction in dark regions further improves the result compared to the reduced ground truth consisting of eight exposures. We computed the Q_{MOS} for all possible permutations of exposures going from one to seven. For the scene at hand, it is difficult to see a difference between the best result obtained by combining two exposures and the resulting radiance maps obtained using more than two exposures. Adding more exposures will, however, reduce the variance of the quality of the resulting HDR radiance maps. The temporal stability has been evaluated for a reference scene containing moving objects. A region-based motion-estimator was able to align the exposures and lead to stable video with almost no artifacts. However, it has to be

noted that only simple motion was present in the scene, and that the motion estimation algorithm is likely to go wrong in case of more complicated motion such as moving people.

Both the disparity estimation for the spatial approach and the motion estimation for the temporal approach have been done using a basic, fast block/region-matching method. We can therefore expect that for both modes, the results can be improved by using a more advanced disparity/motion estimator. Nevertheless, this basic approach allowed to pinpoint the main problematic areas, where we have reason to believe that even the most advanced estimator will face problems. One type of problematic areas are *half-clipped* regions. For these regions, there potentially is a lot of texture in the non-clipped view, and no texture or noise in the clipped view. In order to get a consistent stereo pair, we need to guess where to take the information from the other exposure.
Figure 5.1 shows the difference images $(\mathrm{I}_{c_2}^{\Delta t_{11}} - e \cdot \mathrm{I}_{c_2}^{\Delta t_4})$ and $(\mathrm{I}_{c_2}^{\Delta t_{11}} - e \cdot \mathrm{I}_{c_1}^{\Delta t_4})$ for the temporal and spatial approach respectively $(e = 8\sqrt{2})$. This visualizes the alignment of the two exposures, where the darker the pixel, the better aligned the images are.



(a) Temporal approach (b) Spatial approach

Figure 5.1: Difference images between exposures Δt_4 and Δt_{11}. We can see that in the case of the temporal approach the two images are aligned, allowing to copy the pixel information for the sky part from the short to the long exposure. This is not possible in the spatial approach, as the images are not aligned.

For the temporal approach, under the assumption that the clipped region is not moving, one can simply copy the information from the next frame, where the region is not clipped. This is however not possible for the spatial approach, where the unclipped information lies in another viewpoint. While the proposed *iterative disparity propagation* algorithm can improve the results by looking at the neighboring, unclipped regions, the results are generally worse than for the temporal approach. This approach is bound to fail in *half-occluded*, *half-clipped* regions, i.e. parts of the scene that are visible and not clipped in one camera but not visible and clipped in the other. One example of such a region is at the right side of the window of the ground truth scene. For these regions, the missing information would need to be guessed.

Applying HDR to a multi-view setup combines two research domains that are on its own still hot topics in research, and both contain problems that are yet to be solved. It is

therefore natural that when combining the two domains, even more issues arise that give pointers for future work.
In this thesis, we focused on one carefully chosen HDR scene, which allowed to compare the results. But in order to be able to get more statistically significant results, the ground truth set would need to be greatly extended. It would also be interesting to perform a subjective quality evaluation of the results. Ideally, one would have a stereoscopic HDR display, as there is no objective quality metric that evaluates stereoscopic HDR content.
In order to combine the advantages of the temporal and the spatial approach, an interesting track to follow is to combine the two approaches proposed in this thesis into a *hybrid approach*. This way, the high quality of the radiance maps obtained using the temporal approach could be combined with the higher temporal sampling frequency of the spatial approach. This becomes interesting if the number of exposures in the temporal approach increase. This idea has been proposed by Ramachandran *et al.* [31], but important details on how the spatial approach has been implemented, as well as how the HDR radiance maps are created, are left out. Also, their work was focused on creating HDR images and not videos.
Another interesting direction is to add a third camera to the setup, as outlined in Section 2.3. The left and the right camera would be set to the same exposure time, allowing optimal disparity estimation between the two views. In particular, *half-clipping* would not arise. The third camera, placed in between the two cameras of the traditional stereo setup would be set to another exposure time, or possibly be set to the temporal HDR mode. Knowing the disparities between the left and the right camera, it becomes quite easy to find the corresponding locations in the middle camera. The nice thing about this setup is that HDR can easily be switched off and one has a fall-back to LDR stereoscopic footage, which is not the case for the temporal and spatial stereoscopic HDR proposed in this thesis.

The ambitious aim to implement a stereoscopic HDR pipeline has led to much insight into the problem of producing stereoscopic HDR content, and enabled to pinpoint the most problematic parts in the generation of stereoscopic HDR content.

6

References

[1] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *In International Conference on Computer Vision*, pages 489–495, 1999.

[2] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[3] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Evaluation of hdr tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(3):330–349, June 2008.

[4] Computar. Specifications of the lens H0514-MP2. `http://computarganz.com/file.cfm?id=158`. Last accessed: 17/08/2011.

[5] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. *SIGGRAPH 97*, August 1997.

[6] K. Devlin. A review of tone reproduction techniques. Technical Report CSTR-02-005, Department of Computer Science, University of Bristol, November 2002.

[7] R. Duda, P. Hart, and D. Stork. Pattern Classification. pages 548–549, November 2001.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *In CVPR*, pages 261–268, 2004.

[9] M. Gerrits and P. Bekaert. Local stereo matching with segmentation-based outlier rejection. In *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*, page 66, Washington, DC, USA, 2006. IEEE Computer Society.

[10] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H.-P. Seidel, and H. P. A. Lensch. Optimal hdr reconstruction with linear digital cameras. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 215–222, 2010.

[11] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004 CVPR 2004*, (C):74–81, 2004.

[12] IDS. Manual for uEye cameras. `http://www.ids-imaging.de/frontend/files/uEyeManuals/Manual_eng/uEye_Manual/index.html`. Last accessed: 17/08/2011.

[13] IDS. Specifications of the uEye USB UI-2230ME. `http://www.ids-imaging.com/pdfmodule/products_pdf.php?cam_id=56`. Last accessed: 17/08/2011.

[14] K. Jacobs, C. Loscos, and G. Ward. Automatic high-dynamic range image generation for dynamic scenes. *IEEE Computer Graphics and Applications*, 28:84–93, 2008.

[15] G. Jing, C. E. Siong, and D. Rajan. Foreground motion detection by difference-based spatial temporal entropy image. *TENCON 2004. 2004 IEEE Region 10 Conference*, A:379–382 Vol. 1, 2004.

[16] G. M. Johnson. Cares and concerns of cie tc8-08: spatial appearance modeling. In *HDR imaging. SPIE/IS&T Electronic Imaging Conference*, pages 148–156, 2005.

[17] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. *ACM Trans. Graph.*, 22:319–325, July 2003.

[18] Y. Ma and H. Zhang. Detecting motion object by spatio-temporal entropy. *Multimedia and Expo, IEEE International Conference on*, 0:68, 2001.

[19] S. Mangiat and J. Gibson. High dynamic range video with ghost removal. volume 7798 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, August 2010.

[20] S. Mann and R. W. Picard. On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. *Proceedings of IS&T*, 323:422–428, 1995.

[21] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph.*, 30(4), 2011.

[22] L. Meylan, D. Alleysson, and S. Süsstrunk. A model of retinal local adaptation for the tone mapping of color filter array images. *Journal of the Optical Society of America A*, 24:2807–2816, 2007.

[23] Microsoft. MSDN entry on semaphores. `http://msdn.microsoft.com/en-us/library/ms685129%28v=VS.85%29.aspx`. Last accessed: 17/08/2011.

[24] T. Mitsunaga and S. K. Nayar. Radiometric self calibration. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:374–380, Jun 1999.

[25] N. Sun, H. Mansour, and R. Ward. HDR image construction from multi-exposed stereo LDR images. In *2010 Proceedings of 17th IEEE International Conference on Image Processing (ICIP 2010)*, pages 2973–6. IEEE, 2010.

[26] K.-I. Naka and W. A. Rushton. S-potentials from luminosity units in the retina of fish (cyprinidae). *Physiology 185(3)*, pages 587–599, 1966.

[27] C. Oliver and S. Quegan. *Understanding synthetic aperture radar images*. Artech House remote sensing library. Artech House, 1998.

[28] Omnivision. Omnipixel3-htsm. `http://www.ovt.com/technologies/technology.php?TID=3`. Last accessed: 17/08/2011.

[29] S. Paris, P. Kornprobst, and J. Tumblin. *Bilateral Filtering: Theory and Applications*. Now Publishers Inc., Hanover, MA, USA, 2009.

[30] N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, and V. Lukin. Metrics Performance Comparison for Color Image Database. In *4th international workshop on video processing and quality metrics for consumer electronics*, 2009.

[31] V. Ramachandra, M. Zwicker, and T. Nguyen. Hdr imaging from differently exposed multiview videos. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 85–88. IEEE Computer Society, 2008.

[32] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21:267–276, July 2002.

[33] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, December 2005.

[34] S. Wu, S. Xie, S. Rahardja, and Z. Li. A robust and fast anti-ghosting algorithm for high dynamic range imaging. In *2010 Proceedings of 17th IEEE International Conference on Image Processing (ICIP 2010)*, pages 397–400. IEEE, 2010.

[35] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[36] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet - sRGB. `http://www.w3.org/Graphics/Color/sRGB`. Last accessed: 17/08/2011.

[37] R. Szeliski. *Computer Vision : Algorithms and Applications*. Springer, 2010.

[38] D. Tamburrino, D. Alleysson, L. Meylan, and S. Süsstrunk. Digital Camera Workflow for High Dynamic Range Images Using a Model of Retinal Processing. In *IS&T/SPIE Electronic Imaging: Digital Photography IV*, volume 6817, 2008.

[39] A. Tomaszewska and R. Mantiuk. Image registration for multi-exposure high dynamic range image acquisition. *International Conference in Central Europe on Computer Graphics and Visualization*, 2007.

[40] A. Troccoli, S. B. Kang, and S. Seitz. Multi-view multi-exposure stereo. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 861–868, Washington, DC, USA, 2006. IEEE Computer Society.

[41] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of the ccd imaging process. In *ICCV'01*, pages 480–487, 2001.

[42] C. Varekamp. Method and apparatus for removing false edges from a segmented image. In *International Patent Application WO2004/051 573*, 2004.

[43] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[44] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image oversegmentation. *International Journal of Computer Vision*, 75:49–65, October 2007.